Tyabandha Journal of Arts and Science

ISSN 1744–9669

Volume 2, Number 1

January 2005

Edited by Vaen Sryayudhya

England
25th June, 2007

Distribution
Internet-searchable world-wide

The story of Andromeda †

Kittisak Tiyapan

Andromeda casts down her sigh
and Vega lights my way.

from a song by John Denver

After arriving in Tokyo from England in October, I have found myself studying an intensive Japanese course at the Tokyo Institute of Technology. The course is interesting as well as well organised. We often have study trips on weekends too, for example we went to Kamakura which used to be a prefecture of Japan before Tokyo became a capital of the country.

On this first of November we had another one of such trips again. This time we went to the Tsukuba Expo Centre. Among other things there was one planetarium. We went inside, and this was the story that we listened to on that day. It was from Greek mythology.

One of the stories from Greek mythology which is being referred to very often in European literature, is that of Andromeda. Anyone who reads English literature will surely have come across the story.

Born a daughter to Cepheus, who was a king of Ethiopians, and Cassiopeia (Cassiepeia, Cassiope), Andromeda was a very charming maiden. Her mother, being so proud of her, boasted that she was even more beautiful than the sea nymphs Nereids, daughters of the sea-god Nereus. They made a complaint to Poseidon, the God of the sea, who then flooded the land and sent a sea-monster called Cetus. Cepheus, hoping to lessen the fury of Poseidon, fastened Andromeda to the rock on sea-shore. At that moment Peseus, son of Zeus and Danae, bringing back the head of Medusa whom he had killed, came by. He fell in love with Andromeda and asked Cepheus for her hand. Cepheus said that he would consent to the marriage proposal only if Peseus could get rid of Cetus first, which he did. After killing Cetus he was attacked by an uncle and betrothal of Andromeda, Phineus, whom he then turned into stone by showing his souvenir, the head of Medusa.

Perseus and Andromeda got married and lived together happily. After their death they were turned into constellations in the sky, together with Cepheus, Cassiopeia, and Cetus. Cepheus and Cassiopeia

† Thai translation published in *Sakkayaphab*, **3**, 2, 24–25, November 1995. Association of Thai Professionals in Japan, Japan.

are among the northern constellations; Andromeda, Pegasus, and Cetus are among those of autumn, while Perseus can be seen in the winter sky [2].

In order to give readers some appreciation of the story, we will venture to catch a glimpse at a book by Hugo [1].

> Alas! Will none come to the help of the human soul in this gloom? Is it its destiny forever to await the mind, the liberator, the huge rider of Pegasus and hippogriffs, the aurora-hued combatant who shall descends from the sky with wings, the radiant knight of the future? Shall it always call to its aid the gleaming lance of the ideal in vain? is it condemned to hear the Evil coming terribly through the depth of the abyss, and to see nearer and nearer at hand, under the hideous water, that dragon-head, those jaws reeking with foam, that serpentine waving of claws, distensions, and rings? Must it remain there, with no ray, no hope, abandoned to that horrible approach, vaguely scented by the monster, shuddering, dishevelled, wringing its hands, foreverchained to the rock of night, a sombre Andromeda white and naked in the darkness?

Here Hugo used Andromeda as a metaphor for prisoners of the galleys during the tumultuous period that followed the French Revolution, many of whom were unjustly sentenced.

§ References

[1] Victor Hugo. *Les Misérables.* Wordsworth Editions Limited. 1994. (also another translated version can be found at `gopher://gopher.etext.ort/11/Gutenberg/etext94/lesms10.txt.gz`)

[2] Stuart J. Inglis. *Planets, stars, and galaxies.* John Wiley & Sons, Inc. 1976.

First Year Report

Kittisak Nui Tiyapan [†]

Introduction

Geometry and mathematics

The simplest geometrical figure is a circle while the simplest of all polygons is a triangle. The degree of freedom of triangles increases from the equilateral to the isosceles and the right triangles to the scalene triangles. While spending the summer of 1990 in a traineeship through AIESEC I was introduced to a geometrical puzzle which, as I came to learn later, is called the *flexatube*, but which I conjectured at the time that was from ancient China. This puzzle is made up of sixteen right isosceles triangles tiled into four squares, each comprising of four triangles, which are in turn joined together to form a loop. It can be easily made up using some hard papers, a pair of scissors and cello tape. There are in total twenty hinges, four of which are as long as the hypotenuse while the other sixteen have their length equal to the shorter side of the triangle. By turning these rigid triangles upon their hinges the inside surface of the strip can become the outside and vice versa. I found one solution a week later and back in Thailand a friend of mine found another solution. I think that these two are the only possible solutions but have not been able to proof it.

Having this interest in geometrical puzzles I was delighted at the time to find that the symbol for the men's toilets in Budapest is an equilateral triangle, while that for women's toilets is a circle. No other things beside these are written. Obviously these two geometrical forms suffice and are fully understood by the whole population.

In general dimension we talk about spheres. A sphere in d dimensions has its volume, V, proportional to r^d and its surface, A, to r^{d-1}, so that $V \propto A^{d/(d-1)}$.

The area of sphere in three dimensions is $A = 4\pi r^2$ and the volume $V = \frac{4}{3}\pi r^3$. When $V = 1$, $r = -(-3/\pi)^{1/3}/2^{2/3}$, $(3/\pi)^{1/3}/2^{2/3}$, or $(-1)^{2/3}(3/\pi)^{1/3}/2^{2/3}$ which are numerically $-0.310175 - 0.537239\imath$, 0.62035, -0.310175, $-0.310175 + 0.537239\imath$ in that order. Therefore $\alpha = A|_{V=1} = 6^{2/3}\sqrt[3]{\pi} = 4.83598$

[†] Written while the author was at Chemical Engineering, UMIST, Supervisor, Professor Graham Arthur Davies, The University of Manchester, 7th October 2002

When $V = 8$, $r = (6/\pi)^{1/3}$ and therefore $A = 46^{2/3}\pi^{1/3} = 19.3439$. In order to find α, the surface area per unit volume, one divide the area by $V^{2/3}$, in other words $\alpha = V^{1/3} \cdot A/V = A/V^{2/3}$.

The same is true for other polyhedra. For example in a tetrahedron where x is the length of the side, the vertices can be

$$(0,0,0),(x,0,0),\left(\frac{x}{2},\frac{\sqrt{3}}{2}x,0\right),\left(\frac{x}{2},\frac{\sqrt{3}}{6}x,\frac{1}{2}\sqrt{\frac{93}{35}}x\right)$$

When $V = 1$, one can obtain x by solving the equation

$$1 = \frac{1}{6}\,\text{abs}\left(\begin{vmatrix} 0 & 0 & 0 & 1 \\ x & 0 & 0 & 1 \\ \frac{x}{2} & \frac{\sqrt{3}}{2}x & 0 & 1 \\ \frac{x}{2} & \frac{\sqrt{3}}{6}x & \frac{1}{2}\sqrt{\frac{93}{35}}x & 1 \end{vmatrix}\right).$$

This gives $x = 2\left(\frac{35}{31}\right)^{\frac{1}{6}} = 2.0409$ as the only real positive answer. When x is doubled, V increases from 1 to 8, which means that one would be dividing A by $V^{\frac{2}{3}}$ to obtain α.

The perimeter of a triangle is $s = 3d_e$ and the area $A = (\sqrt{3}/4)l^2$. When $A = 1$, $d_e = \pm 2/3^{1/4} = \pm 1.51967$. Therefore $s = 4.55901$.

Vertices shared by two cells make up a common face between them. Two way have been tried for finding the edges. The first one was by looking at all neighbouring cells of every cell in turn three at a time. The edges are then made up of those vertices that are common among these three cells. Only those edges which have exactly two vertices are considered. They are called *good edges* as contrasted with edges on the boundary. This is a much longer way than the second one, which is to consider vertices common to any two faces of a cell. Similar to the first case, such vertices forms a good edge if and only if there are only two of them. The two methods above give exactly the same list of edges, so they confirm each other. It has been tested that all edges having more than two vertices are boundary ones, that is they have at least one vertex outside the boundary of the unit cube considered.

By drawing some of the cells as a solid using *fill* command it has been tested that the result from `convhull` covers the entire cell surface. This confirms the step where areas are calculated.

The hexagon or honeycomb is perhaps the pattern which is most frequently found in nature. Even though the world we live in is three-dimensional, cells normally divide and spread in two dimensions in the form of layers. Moreover, they are packed in these layers in patterns which most often resemble the honeycomb (*cf* Williams and Bjerknes, 1972).

An octagon is an eight-sided polygon. It is the shape of the cross section of every chimney in the mills built in Manchester during its industrial era of the nineteenth century, as well as that of the turrets in the Main Building of UMIST. Perhaps one of the reasons for its popularity is that it looks strong while having the style of a good taste. May be the reason why it looks strong is that it possesses eight axes of symmetry, on top of another symmetry around the origin.

There are nine regular polyhedra. Among these are five regular convex solids known to the ancient Greek called Platonic polyhedra. They are tetrahedron, cube, dodecahedron, octahedron, and icosahedron. They have regular congruent faces and regular polyhedral angle vertices. Their face angles and their dihedral angles at every vertex are equal. The other four regular polyhedra have only been discovered much later and are not convex. They are called the Kepler-Poinsot polyhedra and are nonconvex. The small stellated dodecahedron and the great stellated dodecahedron were found by Kepler (1571–1630). The great icosahedron and the great dodecahedron were found by Poinsot (1777–1859). The small stellated dodecahedron and the great dodecahedron do not satisfy Euler's equation. The process of creating it by extending nonadjacent faces until they meet is called *stellating*. There are also polyhedra called quasi-regular.

The semi-regular polyhedra are called the Archimedean polyhedra. Here all faces are regular polygons but not all are of the same kind. Every vertex is congruent to all others. They comprise of an infinite group of prisms, an infinite group of antiprisms or prismoid, and another thirteen polyhedra. Each prism or prismoid is made up of two regular polygons on parallel planes where the vertices are aligned in the former case or shifted half way to the next neighbouring vertices in the latter case. Each vertex in prisms is joined to a corresponding vertex of the opposite polygon, while in prismoid it is joined to two corresponding vertices. All faces of an Archimedean solid are regular and all its polyhedral angle vertices congruent.

On the other hand the Archimedean duals have the property that all their faces are congruent to one another and all their polyhedral angles regular. These solids are important in crystallography. They are vertically regular and include an infinite group of dipyramids, an infinite group of trapezohedra, and additionally thirteen other polyhedra.

The surface area per unit volume α of a solid can be computed from the actual volume V and the actual surface area A as $\alpha = V^{1/3}A/V = AV^{-2/3}$.

Polygon	n_e	A	s (numerical)]
Triangle	3	$(\sqrt{3}/4)d^2$	4.55901
Square	4	d^2	4
Pentagon	5	$5(1+\sqrt{5})d^2/\left[4(10-2\sqrt{5})^{1/2}\right]$	3.8119
Hexagon	6	$3\sqrt{3}d^2/2$	3.7224
Heptagon	7	$(7/4)d^2\tan(5\pi/14)$	3.6721
Octagon	8	$2d^2\tan(3\pi/8)$	3.6407
Nonagon	9	$(9/4)d^2\tan(7\pi/18)$	3.6198
Decagon	10	$5\left[(5+\sqrt{5})/2\right]^{1/2}d^2/(-1+\sqrt{5})$	0.3605
Undecagon	11	$(11/4)d^2\tan(9\pi/22)$	3.5944
Dodecagon	12	$3(2+\sqrt{3})d^2$	3.5863

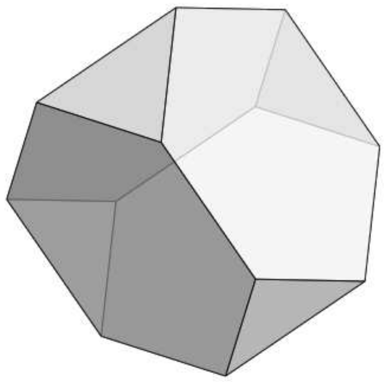
Table 1 *Perimeter per unit area of n-gons.*

The tetrahedron is self-dual. The octahedron is dual to the cube while the dodecahedron the icosahedron.

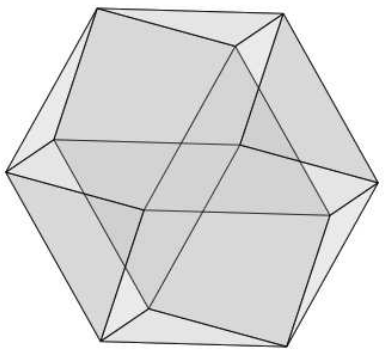
The nearest neighbour and minimum spanning tree have been applied to the problem of taxonomy in botany. Clayton (1972), working on the characters of plants to manually classify them (*eg* Clayton, 1970) with the use of only the binary dendrogram and trial and error, adopted a numerical method which finds the minimum spanning tree in a multi-dimensional character space. Since taxonomy can be considered as a kind of dictionary, it is possible to apply a similar approach to machine translation and the compilation of dictionaries.

The icosahedron is common shape found among viruses.

The polyhedra from Figure 1 to 3 are semi-regular.



(a)



(b)

Figure 1 (a) *Truncated tetrahedron, triakistetrahedron,* 2 3|3. (b) *Octahemioctahedron, octahemioctacron,* 3/2 3|3

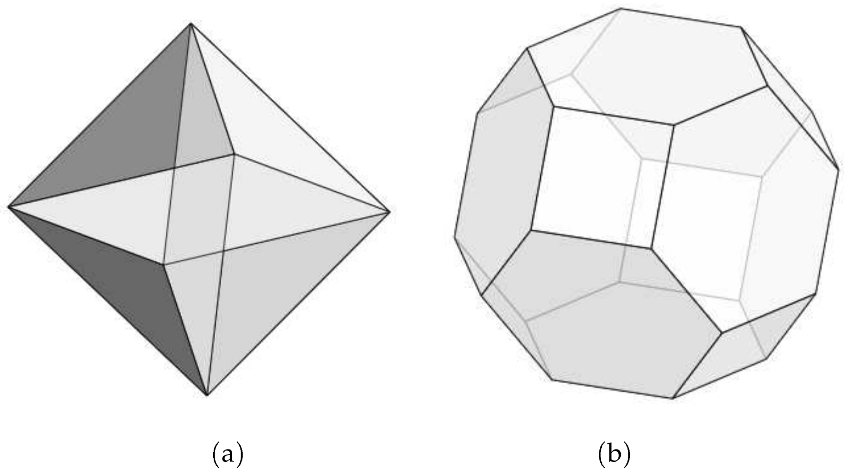


Figure 2 (a) *Tetrahemihexahedron, tetrahemihexacron,* 3/2 3|2. (b) *Truncated octahedron, tetrakishexahedron,* 2 4|3

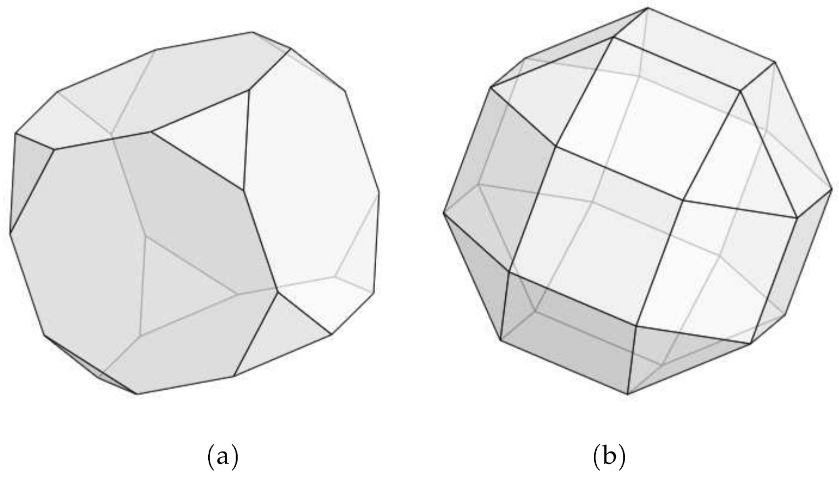


Figure 3 (a) *Truncated cube, triakisoctrahedron,* 2 3|4. (b) *Rhombicuboctahedron, deltoidal icositetrahedron,* 3 4|2

Polyhedra in Figure 4 are snub polyhedra.

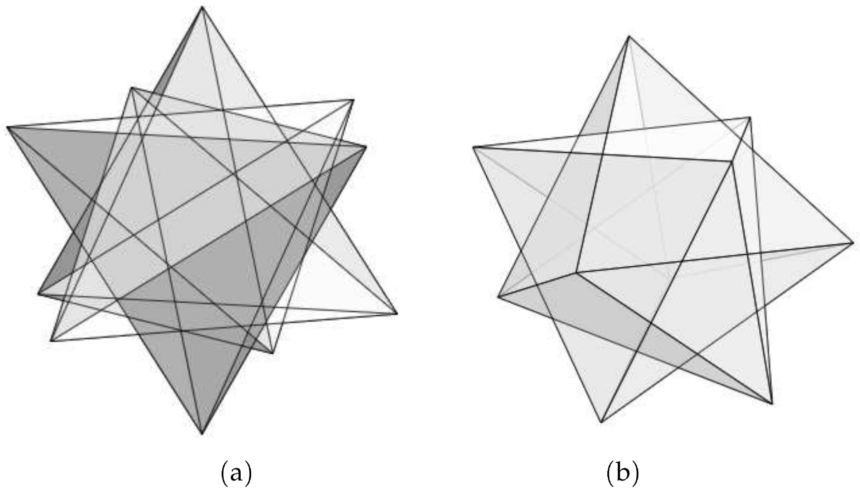


Figure 4 (a) *Pentagrammic crossed antiprism, pentagrammic concave deltohedron,* |2 2 5/3. (b) *Pentagrammic antiprism, pentagrammic deltohedron,* |2 2 5/2

The surface of Fullerine is made up of pentagons six-sided figures. Its shape represents that of the geodesic domes developed by Buckminster Fuller, and hence the name *Fullerine*. The latter may either be hexagons or figures all the six sides in each one of which form two sets of three sides having an equal length. The simplest Fullerine, the carbon-60 molecule, has the same shape as that of a football and a handball. With some thought the reason for this is not difficult to see. With its thirty-two faces it closely resemble the sphere. Also the two different shapes of all its components are symmetrically distributed and therefore enable colouring with only two different colours, namely one for each of the two shapes. To see how this helps, suppose one made a football in the shape of a bloated dodecahedron. Then it would be impossible to colour it using more than one colour at the same time of giving it a symmetrical appearance when viewed from more than a few directions. With the Fullerine shape and the colouring scheme mentioned, however, the football looks symmetrical when viewed from 54 different directions symmetrically distributed around it. These directions corresponds to those when one looks at it in the direction perpendicular to the centre of each of its faces and when in the direction through the middle of each of the 22 edges lying between two hexagonal faces.

Making polyhedron models is an educating experience. Contrary to the general believe that you need to make an accurate drawing for the required parts (Wenninger, 1971), this needs not be so. Examples of this are the origami models of polyhedra where complex polyhedron structures are made from interlocking pieces each of which is made by folding a piece of paper of a rectangular or square shape.

A set of elements with the sum and the product of any two elements defined is a commutative ring if under these two operations it satisfies the following postulates: closure, uniqueness, commutative, associative, and distributive laws, identity (zero and unity), and additive inverse. An integral domain is an ordered domain if its positive elements satisfy the laws of addition, multiplication, and trichotomy. A subset of an ordered domain is well-ordered if every nonempty subset of it contains a smallest member. $a|b$ means that $.b$ is divisible by a. The Euclidean algorithm or division algorithm states that $a = bq + r, 0 \leq r < b$. Two integers are relatively prime if their only common divisors are ± 1. $a \equiv b (\operatorname{mod} m)$ if and only if $m|(a-b)$. The commutative ring Z_2 is the properties of multiplication and addition of even (0) and odd (1) numbers.

$+$	**0**	**1**
0	0	1
1	1	0

$\cdot$	**0**	**1**
0	0	0
1	0	1

The following is Z_5.

$+$	**0**	**1**	**2**	**3**	**4**
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

$\cdot$	**0**	**1**	**2**	**3**	**4**
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

There is a close link between geometry and algebra. Geometrical surfaces can be described as algebraical equations. For example, for circles and polygons the equations are binary quadratic, while for spheres and polyhedra they are ternary quadratic. Even one-sided surfaces can be described algebraically. The equation of Klein bottle, when deformed into a sphere with two circles removed and replaced by two cross-caps, is a quartic equation

$$a^2(x^2+y^2)(b^2-x^2-y^2) = z^2(a^2x^2+b^2y^2),$$

while the Steiner surface is also a quartic one

$$y^2z^2 + z^2x^2 + x^2y^2 + xyz = 0.$$

Two surfaces is homomorphic to each other if it is possible to continuously transform one into the other. All convex polyhedra are homomorphic to a sphere. The Steiner surface is homomorphic to the heptahedron, which is an Archimedean polyhedron with diametral plane.

In the plane, a second-degree equation gives either two straight lines, a circle, an ellipse, a parabola, or a hyperbola. In space, it can give two planes, cylinders and cones (circular, elliptic, parabolic, or hyperbolic), sphere, spheroid, ellipsoid, two hyperboloids, and (elliptic or hyperbolic) paraboloid.

Partition, *tessellation* and *division of space* are the same thing. In the context of set theory, a partition of set X is a family of sets A_1, A_2, …, A_k which are subsets of X, such that $A_i \neq \emptyset$; $A_i \cap A_j = \emptyset$; $\bigcup_i A_i = X$, where i, $j = 1, 2, \ldots, k$ and $i \neq j$. (*cf* Berge, 1958) A further condition that makes any tessellation a Voronoi one is that, for all i there exists a unique point a_i within A_i such that every point in A_i is closer to a_i than to any other a_j, $j \neq i$.

Voronoi tessellation in three dimensions can be constructed by imagining each region as a spherical cell growing outwards to meet neighbouring cells and continue growing to fill the gaps. The centre of each sphere is a unique nucleus point of the region such that it is closest to any point belonging to that region than any nuclei points. If the rate of growth is the same from every cell, the resulting partitions will be planes which can be described by ternary quadratic equations. However, if this rate differs from one cell to another, the partitions will be curved surfaces and the result is a non-Voronoi tessellation. It is possible to impose a constraint of minimum distance between neighbouring nuclei. Such cases can be looked at as spheres of an equal nonzero radius expanding away from nucleus centre points. If the radii differ from one sphere to another, or if some nonspherical solids are used instead of spheres, the tessellation obtained will be non-Voronoi.

Consider the case where all spheres are of equal size. If these spheres already touch their neighbours before the expanding starts, the case is that of packed spheres expanded to form a Voronoi tessellation. There are two types of close-packing: cubic (face-centred) and hexagonal. In both cases each sphere has twelve neighbours. Both cases have the same density, which is $\frac{\pi}{3\sqrt{2}}$. The Voronoi regions produced from the cubic case are rhombic dodecahedra and the faces are rhombuses. In the case of hexagonal close-packing, the corresponding regions are trapezo-rhombic dodecahedra and the faces are either rhombics or trapezia. Where the spheres meet with their three neighbours in the layer above and their three neighbours in the layer below, the faces are rhombics. Where they meet with the six neighbours on the same layer they are trapezia.

For geometrical calculation, an example of a definitive book is that written by Salmon (1912).

The gamma function,

$$\Gamma(z) = \int_0^\infty e^{-t} t^{z-1} \mathrm{d}t, \quad \mathrm{Re}(z) > 0, \qquad (\mathbf{1})$$

got its name from Legendre and is known as the Euler gamma function or simply the second Euler function. The formula $\Gamma(z+1) = z\Gamma(z) = z!$ recursively calculates the gamma function from, for instance, $\Gamma(1/5) \approx 4.5908$, $\Gamma(1/4) \approx 3.6256$, $\Gamma(1/3) \approx 2.6789$, $\Gamma(2/5) \approx 2.2182$, $\Gamma(1/2) = \sqrt{\pi} \approx 1.7725$, $\Gamma(3/5) \approx 1.4892$, $\Gamma(2/3) \approx 1.3541$, $\Gamma(3/4) \approx 1.2254$, and $\Gamma(4/5) \approx 1.1642$. The Stirling's formula was found by de Moivre which approximates the gamma function. The gamma function expansions is

$$\Gamma(x+1) = \lim_{k\to\infty} \frac{k^x 1\cdot 2\cdot 3\cdots k}{(x+1)(x+2)\cdots(x+k)}, \tag{2}$$

and the gamma function of negative numbers can be obtained from

$$\Gamma(-z) = \frac{-\pi}{z\Gamma(z)\sin \pi z}. \tag{3}$$

The incomplete gamma function is

$$\Gamma(z,x) = \int_0^x e^{-t}t^{z-1}\mathrm{d}t = \int_x^\infty e^{-t}t^{z-1}\mathrm{d}t, \tag{4}$$

and the normalised or regularised incomplete gamma function is

$$\frac{\Gamma(z,x)}{\Gamma(z)}$$

Statistics

Poisson distribution, defined by $p(x,\lambda) = \left[\lambda^x e^{-\lambda}/x!\right] \mathrm{I}_{[0,\infty)}$, is the binomial distribution, $p(x,n,p) = {}^n\mathrm{C}_x \theta^x (1-\theta)^{n-x}\mathrm{I}_{[0,n]}$, when n goes to infinity, θ goes to zero, while $n\theta = \lambda$. Here θ is the probability of success of each trial. It is used when counting the number of occurrences of a random event. Analogously Poisson point process, which has $p(x = n(v)) = \left[\lambda|v|e^{-\lambda|v|x}/x!\right] \mathrm{I}_{[0,\infty)}$,, is the binomial point process, $p(x = n(v)) = {}^n\mathrm{C}_x \theta^x (1-\theta)^{n-x}\mathrm{I}_{[0,n]}$, when the volume V goes to infinity, while $n/|V| = \lambda$. Here $\theta = |v|/|V|$ is the probability of points within V being placed in $v \subset V \subset R^d$, and λ the density or intensity of points. Therefore the density of point of a Poisson point process is constant by definition. A point process is a procedure which generates points on a domain within a space of d dimensions.

The Poisson point process thus derived has the properties that $0 < p_{n(v)=0} < 1$ for $0 < |v| < \infty$, $\lim_{|v|\to 0} p(n(v) \geq 1) = 0$, $n(v_i)$ mutually independent and $n(\bigcup_n v_i) = \sum_n n(v_i)$ when A_i are disjoint, and $\lim_{|v|\to 0} \left[p(n(v) \geq 1)/p(n(v) = 1)\right] = 1$.

The weighted mean of a group of data is $\mathbf{x} = \sum_i f_i x_i/n$ and the weighted variance is $\sigma^2 = \sum_i f_i(x_i - \mathbf{x})^2/n$, where f_i is the occurrence frequency of x_i and $\sum_i f_i = n$. Likewise the r^{th}-moment around the average is $m_r = \sum_i f_i(x-\mathbf{x})^r/n$, while the r^{th}-moment around the origin is $m'_r = \sum_i f_i x^r/n$ (*cf* Spiegel, 1975). Some relations among these various moments are $m_1 = 0$, $m_2 = m'_2 - {m'_1}^2$, $m_3 = m'_3 - 3m'_1 m'_2 + 2{m'_1}^2$, and $m_4 = m'_4 - 4m'_1 m'_3 + 6{m'_1}^2 m'_2 - 3{m'_1}^4$.

The variance when normalised by $n-1$ gives the best unbiased estimated variance if the sample has a normal distribution. On the other hand the variance which is normalised by n is identical with the second moment of the sample about its mean.

Phase transition

In the Ising model each spin has two possible states, that is up and down, and the Hamiltonian is $H = J_0 \sum_{<i,j>} \sigma_i \sigma_j$ where the summation is over the nearest neighbours. Since it has been exactly solved, the Ising model provides a good model for the understanding of phase transition. This model can represent the transition from ferro- to paramagnetic at the critical temperature where the correlation length becomes infinite. Characteristic to the Ising model is the peak in the specific heat at the critical temperature.

The two-dimensional *xy* model is a model of spins confined to a plane, the Hamiltonian of which is $H = J_0 \sum_{<i,j>} \cos(\theta_i - \theta_j)$. This model can represent the superconducting and the superfluid films. For this model there is no phase transition showing long-range ordering. One example is the two-dimensional Coulomb gas model where the vortex-antivortex pairs, which are bound to each other at low temperature, increases in number as the temperature increases and become separated at the KT temperature that marks the phase transition.

It had been generally believed that no phase transition can exist for the *xy* model when Kosterlitz *et al* (1973) showed that there is another kind of phase transition, arisen from the topological excitation of vortex-antivortex pairs instead of from the long-range ordering found in a spontaneous magnetisation. They consider the two-dimensional model of gas with charges $\pm q$ where the interaction potential is $U(|\mathbf{r}_i - \mathbf{r}_j|)$ is $-2q_i q_j \ln|(\mathbf{r}_i - \mathbf{r}_j)/\mathbf{r}_0| + 2\mu$ when $r > r_0$, and 0 when $r < r_0$. The problem is reduced to that of solving an equation of the form $(\mathrm{d}y/\mathrm{d}x) = -e^{-xy}$. The application mentioned there is in the *xy* model of magnetism, the solid-liquid transition, and the neutral superfluid, but not in a superconductor and a Heisenberg ferromagnet.

The frustrated *xy* model, the Hamiltonian of which is

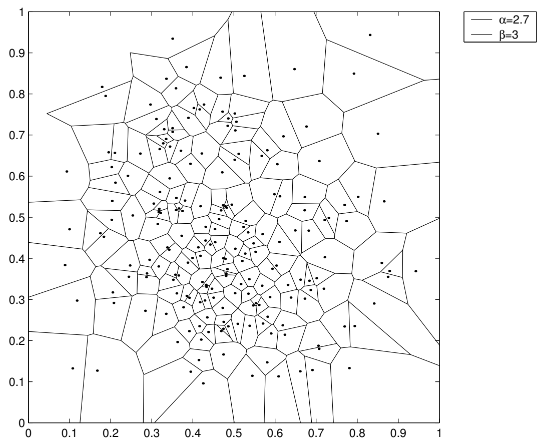
$$H = J_0 \sum_{<i,j>} \cos(\theta_i - \theta_j - A_{ij})$$

occurs when a magnetic field is applied perpendicular to the two-dimensional plane of the xy model. The frustration parameter, $f = \Phi/Phi_0$, is a measure of the average external magnetic flux. When $f = 1/2$ the model is called the fully frustrated xy model. The local chirality, $m(r_i) = \frac{1}{2\pi}\sum(\theta_i - \theta_j - A_{ij})$, which describes the property of the ground state, where it can either be $+1/ovr2$ or $-1/ovr2$. The network configuration at T < T$_c$ is that of a draught board, and has Z$_2$ symmetry. This regularity is broken by the formation of domain walls in an Ising phase transition at T$_c$.

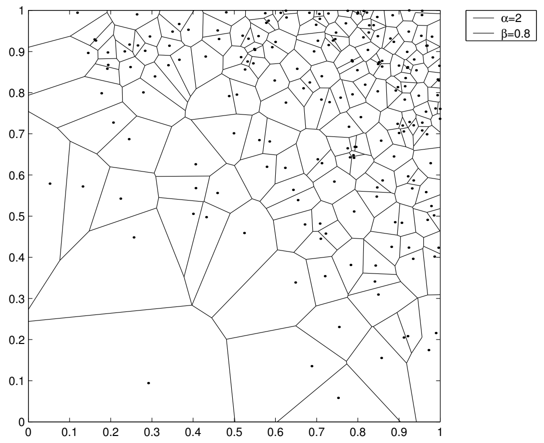
Random processes

A synonym to *random* is *stochastic* (*cf* Miles, 1972). Any algorithm which employs a random element is called Monte Carlo. Random processes can have various types of distribution. The beta distribution has a probability density function $f_{X_{\alpha,\beta}}(x) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha,\beta)}$, $0 \leq x \leq 1$, where $\alpha > 0$ and $\beta > 0$ are shape parameters, and $B(\alpha, \beta)$ is the beta function. There are three types of shape; the bridge shape has $\alpha > 1$ and $\beta > 1$, the J shape $\alpha \leq 1$ and $\beta \geq 1$, or $\alpha \geq 1$ and $\beta \leq 1$, and the U shape $\alpha < 1$ and $\beta < 1$.

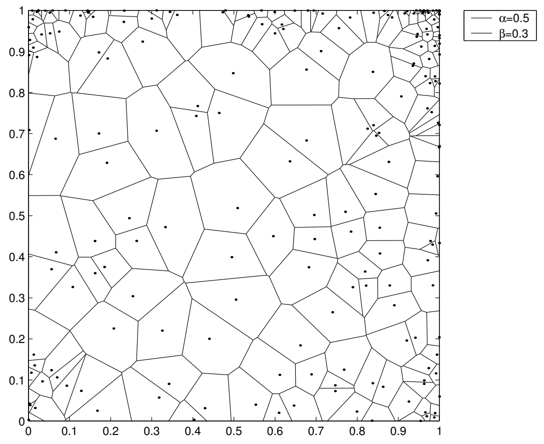
The 200 generators used in Figure 5 are randomly chosen with beta distribution with the shape parameters $\alpha = 2.7$ and $\beta = 3$, that is bridge shape.



Both x and y in Figure 6 have J-shaped distribution with the shape parameters $\alpha = 2$ and $\beta = 0.8$. The density is unbounded at $x = 1$ and at $y = 1$ because $\beta < 0$ for both.



The shape parameters in Figure 7 are $\alpha = 0.5$ and $\beta = 0.3$, that is U shape. The density is unbounded at $x = 0$, 1 and at $y = 0$, 1 because α is also less than zero.:



Structures in nature

Prusinkiewicz and Lindenmayer (1990) model the structures in plants after having briefly discussed about the difference between the Chomsky grammars and the L-system, both of which being a mean for doing string rewriting, but the latter is based on the Turtle geometry which makes it convenient for the geometric rewriting of fractals. The states of a turtle consists of its position coordinates and the direction in which it is facing. Meinhardt (1995) models the patterns on sea shells by using mathematical based on the partial differential equations governing the system of activator, inhibitor, and substrate. Starting from a homogeneous initial condition, small deviations therein undergo a positive feedback and therefore increase. Activator catalyses both the production of itself and that of its inhibitor. The latter acts as a negative feedback which limits and makes the reaction local.

Random tissue in three dimensions has four edges, six faces and four cells meeting at each vertex. It is thus surrounded by four cell nuclei, as well as by six bonds forming a tetrahedral cage. The four edges meeting at a vertex resemble a caltrop, and similarly the lines from it to the four nuclei. If a vertex had more than four edges it would have been structurally unstable, because then it can be split into two normal vertices by an infinitesimal deformation. Continuous random networks, for example models of covalent glasses like vitreous silica, are excluded from this restricted class since theirs may be more than three non-planar faces to an edge even if they still have four edges to a vertex (Revier, 1982).

Unlike crystallography, the ideal random structure is by no means unique because it is the solution of a statistical problem. There are, however, certain geometrical and topological invariances, the most famous of which is possi bly the Euler's theorem. In two dimensions this theorem states that $f-e+v=\chi$, where χ is an Euler-Poincaré characteristic and integer of order one, χ being for instance 1 and 2 respectively for plane and sphere; in three dimensions it is $f-e+v=2$. The valence relations, $\sum nf_n = 2e = 3v$, hold for 2-d and 3-d alike.

Stumbling upon some observations, Theorem's 4 and 5 resulted. These two theorems help explain together with Algorithm ℵ on page ℵ, the valence relations. Theorem 1 is also another product obtainable from applying both the Euler's theorem and the valence relations (*cf* Prause, 2000).

Theorem 1. The average number of edges per polygon in a large pattern is six.:

Proof.: From Euler's theorem, $f-e+v=1$, and the valence relations, $\sum_n f_n = 2e = 3v$, it follows by applying the latter to the former that $f-\sum nf/2+\sum nf/3=1$. Since $\bar{n}$ is the average number of edges per face, it follows that $f-\bar{n}f/2+\bar{n}f/3=1$. Then $f(1-n/6)=1$, and consequently $1-1/f=n/6$. As the network becomes large, f becomes infinite and as a result $\bar{n}=6$. ▫

It is interesting to note that Theorem 1 posts no restrictions on whether the network is random or regular. The only assumption made is that three and only three edges meet at each vertex. One is almost tempted to say that, as the size of a network becomes infinite, nature somehow follows this theorem and make sure that each of the polygons has six edges on average.

When I translated the three papers by Voronoi (Tiyapan, 2001) I used the term *vertex* to mean a vertex of a specific polygon or polyhedron. For any vertex, I used the term *vertice*, and for more than one vertex vertices. It turns out that I am not the only one who concerns himself with the word. Moore and Angell (1993), for instance, use *apex* for a single corner, *vertex* to mean any point in a tessellation in two or three dimensions where its apices meet.

Rivulets flowing inside a pack bed have been studied by several authors (*cf* Porter, 1968). They are said to flow independently of each other, with no mixing among one another. Diffusion theory has been used to treat a random walk process.

The van der Waals force in combination with double layer repulsion play an important part in the study of filter and particle movements in porous media. Both are electrical forces and can be used to explain the particle capturing mechanism.

Percolation is related to chemical engineering contexts (*cf* Mohanty *et al*, 1982). But the necessary background in stochastic processes for the purpose of simulation has already been described earlier, for example the modelling of colmatage, the retaining of particles suspended in a fluid flowing through a porous medium (Litwiniszyn, 1963 and 1967). In general, however, few authors in all engineering fields relate their works directly to the percolation theory. The majority of studies in this area are based on dynamics and fluid dynamics theories (*cf* Mulder and Gimbel, 1990; Kock and Judd, 1965).

The reason for the lack of percolation material in engineering literature may be because the percolation process generally works behind the scene and only shows itself as critical phenomena. Most engineering studies are concerned with things under some operational condition, within the range of which percolation seems to be absent. By contrast, in Physics where extreme conditions are considered, there is an enormous and increasing amount of publications which are directly under the topic of percolation. But this by no means means that within more pacific ranges no places exist for percolation. In fact it is precisely this lack of mentioning in the literature that induces one to this kind of research. Because our idea of criticality is by tradition closely linked to the idea of time, percolation seems to be present only when there are instantaneous changes. But if in a steep s-curve we only rotate the axis clockwise by $\pi/2$ radian, such that to make the time axis vertical instead of horizontal, then we will see that in place of one critical point in the middle of the graph connecting two different levels, there are now two critical phenomena on both sides, one on each side, and in the middle a flat region where time hardly changes. Looking at it this way percolation seems to be a symmetry between time and space. In physical systems spaces percolates, but in the dual world where criticality is continuity it is the time instead which percolates.

Having said that, without rotating the time-space axes backwards and forwards too often one should still find ways to investigate what a pacified percolation does. In this regard, the study of economics seems to be an ideal place to start, if simply because one knows there exists such thing as hyperinflation but one never wants to study that when it happens. This automatically forces the researcher to find ways of doing researches which would not ruin his pocket or put his life in jeopardy. Another ground with a good prospect is in traffic congestion, even though its worst effect is not yet devastating, apart from what it sometimes does to the economy.

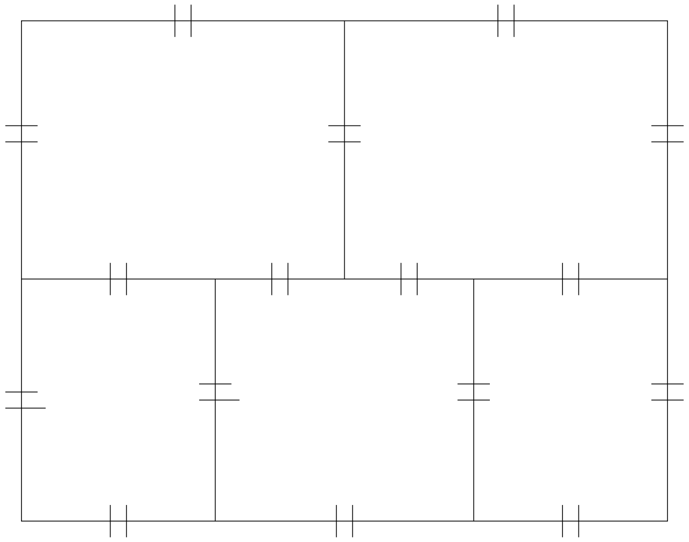
With these digressions in mind, if we now turn our thought at this point to our chemical engineering studies, we will not fail to see how ideal filtration fits the no-ruins requirement. For here we have a process which percolates routinely, often needs to be backflushed, but where the effect it produce is probably that of giving engineers a headache and, at its worst, putting a decent company out of business. But even with this advantage, I still believe that the study of filtration should not concentrate only on the fouling of filters, but should try to understand both the percolated and nonpercolated situations, preferably the latter for the lack of it in literature, and to connect what happens in a working filter to what happens, or does not happen, in a fouled one.

The revolutionising discovery made by F. August Kekulé (Kekulé, 1865, cited in Wotiz, 1993) that benzene has cyclic nature gave rise to the structural theory of organic chemistry.

Voronoi tessellation

In a puzzle of Figure 8 there are five rooms with doors in the position as shown. The problem is whether one can walk through every door only once and the answer according to the graph theory is *no*, because there are more than two rooms which has an odd number of doors. The proof of Theorem 2 was from Komsan Bajȧṛavȧṇijẏ around 1989. From this, when one plays a puzzle like that of Figure 8 one always starts off from a room with an odd number of doors and ends in another such room. This is the same thing as saying that one starts and ends outside rooms with an even number of doors. Therefore the number of the latter is of no consequence, but that of the former is crucial for the existence of a solution and must never be any number other than two.

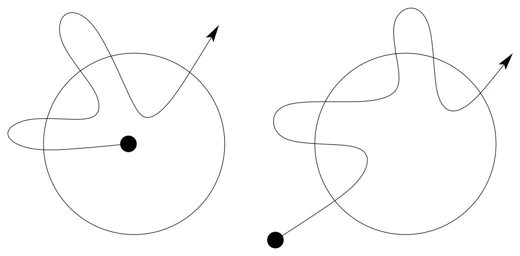
In Figure 8 there are three rooms with five doors, two with four, and one with nine. There are here four rooms with an odd number of doors. Starting off from one of these four one can only end up in one of the other three, which leaves the remaining two rooms unaccounted for. In other words at least two doors will necessarily remain unvisited.



Theorem 2. A travel along a network can only starts and ends at nodes which have an odd coordination number.

The proof of Theorem 2.

Proof.: Looking at Figure 9 if one starts from inside a room with an odd number of doors, one always ends up outside it. On the other hand one always ends up inside a room with an even number of doors if one starts of from it. In the second picture such a room is all the area outside the circle.



□

The following Corollaries 2[1], 2[2] and 2[3] assume nondegeneracy of the Voronoi network. Such a path as mentioned in these corollaries is also called *self-avoiding*.

Corollary 2[1]. There can be no path which traverses all edges of a Voronoi graph only once.

Proof.: This follows from Theorem 2 because a two dimensional Voronoi network has a coordination number three. □

Corollary 2[2]. On a three dimensional Voronoi structure there always exists a path that runs through every edge once and only once.

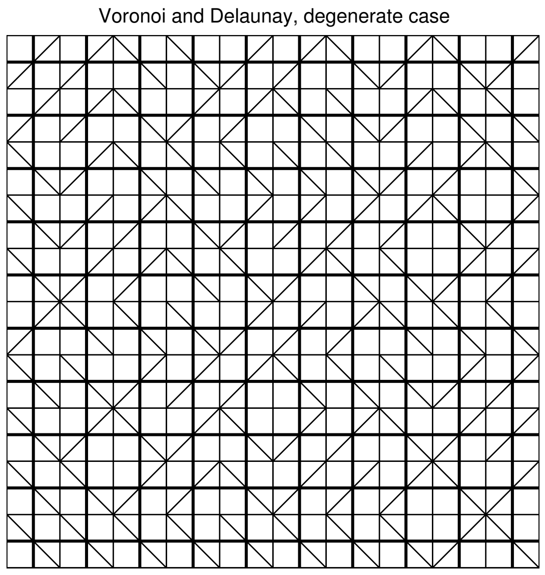
Proof.: This also readily follows from Theorem 2 since a three dimensional Voronoi network has a coordination number of four. ▫

Corollary 2[**3**]**.** Take any Voronoi cell of the three dimensional network, it is impossible to walk through all its edges without repeating some of them.

Proof.: Again from Theorem 2 and because the surface of a Voronoi polyhedron is a two dimensional network of polygons which has the coordination number three. ▫

Jerauld *et al* (1984) compared the Voronoi, with the triangular networks and found that the bond percolation probability of the former is 4.3% or 0.015 less than that of the latter, small site clusters more, and small bond clusters less likely.

When a square lattice was fed to *voronoi* and *delaunay* in Matlab, by the programme `degen.m`, there were error messages saying that points were collinear and possibly triangulation is incorrect. This case, Figure 10, is degenerative.



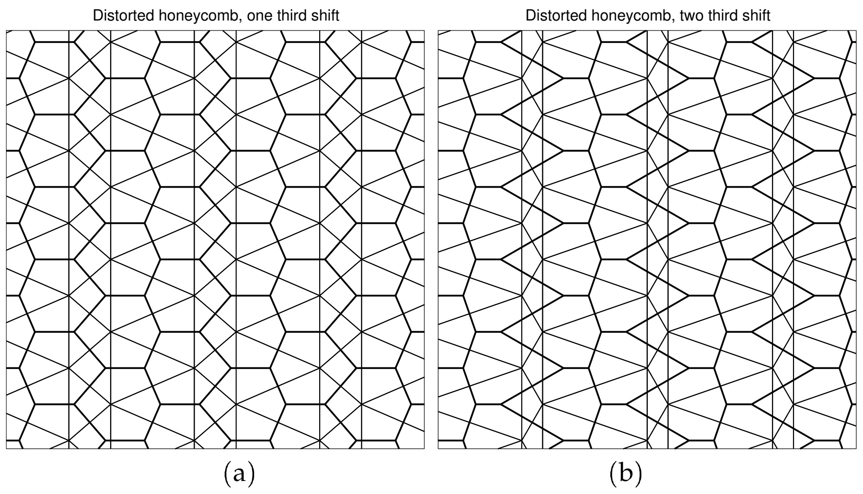

Distorted honeycomb, one third shift
Distorted honeycomb, two third shift
(a)
(b)

Figure 11 *The honeycomb or hexagonal lattice whose alternate y-plane has been shifted (a) one-third, and (b) two-third respectively. Triangulation is shown with thinner lines. The programme used is* `honey.m`

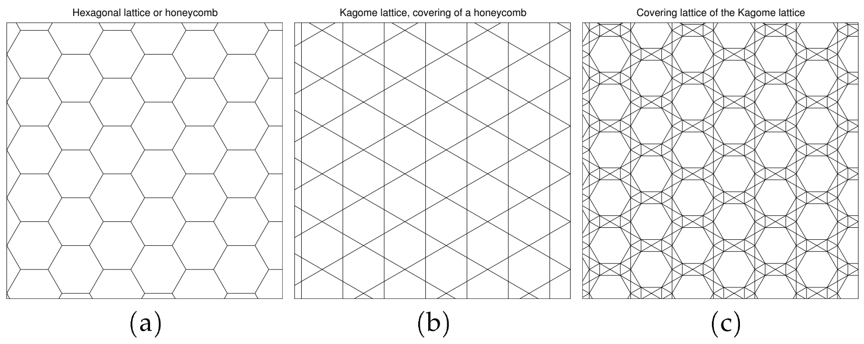
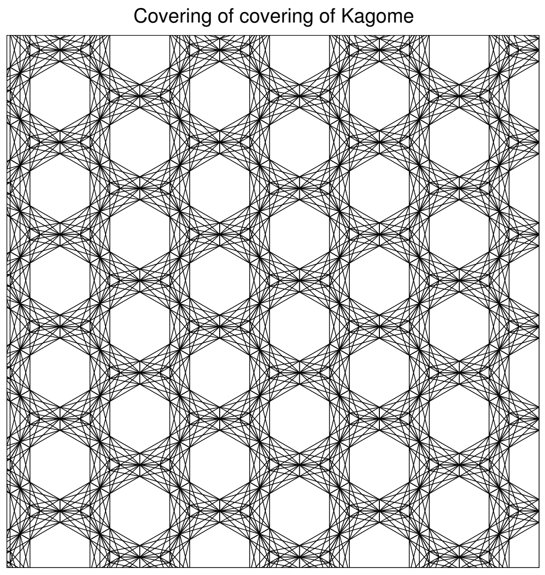
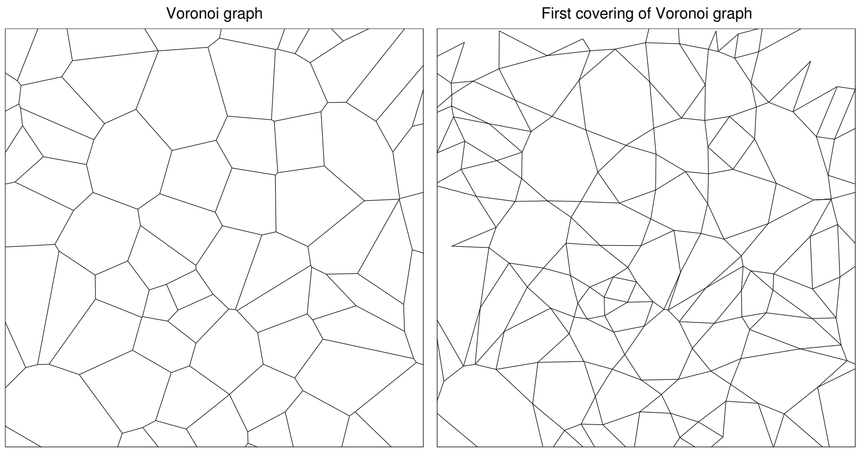

Hexagonal lattice or honeycomb
Kagome lattice, covering of a honeycomb
Covering lattice of the Kagome lattice
(a)
(b)
(c)

Figure 12 (a) *The hexagonal lattice,* (b) *its covering lattice (Kagome), and* (c) *the covering lattice of its covering lattice (i.e. covering of Kagome). (*`cover.m`, `covers.m`*, and* `coverss.m`*)*

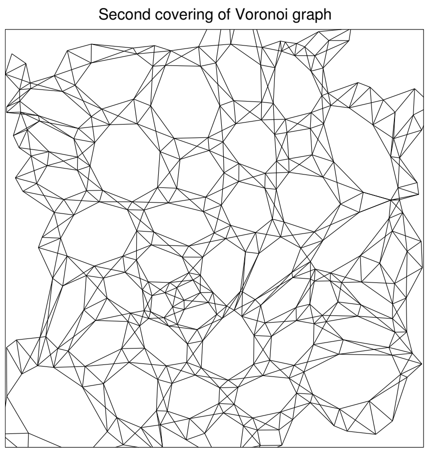
If we indicate by $C_v^n(x)$ the n^{th}-order covering lattice of a lattice x, then the first picture is Hexagonal, the second one Kagome or C_v^1(Hexagonal) and the third one C_v^2(Hexagonal) or in other words C_v^1(Kagome). Figure 13 is the next iteration, a C_v^3(Hexagonal) or C_v^2(Kagome).



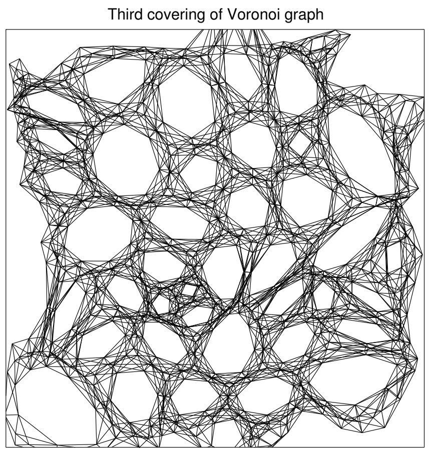
Now let us look at the Voronoi graph and its covering lattices. Pictures in Figure 14 are drawn by first creating and cropping a Voronoi graph with the help of the programme `crop.m`, then use the recursive procedure described above to find up to the third covering lattice.



(a) (b)



(c)



(d)

Figure 14 (a) *Voronoi graph (V.g.),* (b) C_v^1(**V.g.**), (c) C_v^2(**V.g.**), (d) C_v^3(**V.g.**).

Notice that the covering lattices retain the skeleton structure of the original Voronoi graph. Those of higher orders represent closer the structures of nature where walls have thickness.

Quadratic equations

Quadratic equations are equations of binary quadratic forms. Around 400 BC Barbilonia had algorithmic equivalences of quadratic equations which are based on the method of completing the square and where all answers are unsigned, *i.e.* positive, lengths. Because there was no notion for zero, Diophantus considered three types of quadratic equations $ax^2 + bx = c$, $ax^2 = bx + c$, and $ax^2 + c = bx$. Euclid, *circa* 300 BC, used geometric equivalences or quadratic equations whose roots are also lengths. Brahmagupta allowed negative quantities, which he called debts, and used abbreviations for the unknown. Al-Khwarizmi classified quadratics into six types, namely squares equals roots, squares equals numbers, roots equal numbers, squares and roots equal number, squares and numbers equal roots, and roots and numbers equal squares. In his book *Liber embadorum*, published in 1145, Abraham bar Hiyya Ha-Nasi (aka Savasorda) gives the complete solution of quadratic equations. Luca Pacioli published *Summa de arithmetica, geometrica, proportioni et proportionalita* (or Summa) in 1494. He also applied quadratic methods to quartics of the form $x^4 = a + bx^2$. Scipione del Ferro solved the cubic equations of the form $x^3 + mx = n$.

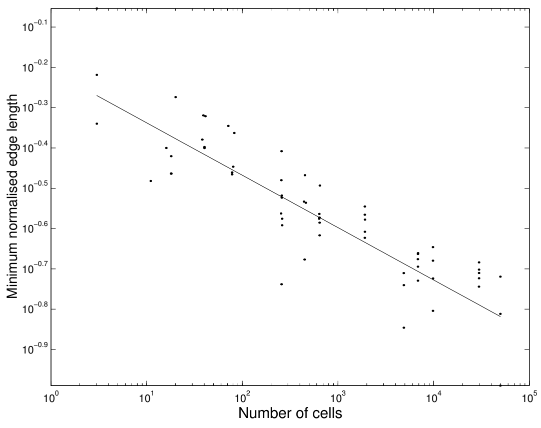
Quadratic forms

The theory of quadratic forms and the theory of matrix are inseparable though the history of these two subjects are somewhat fragmentary. A bilinear form in the sets x_i and y_i, $i = 1 \ldots n$, is $\sum_{i,j} x_i y_j$, or $\mathbf{x}^{\mathrm{T}}\mathbf{Ay}$. If $x_i = y_i$ for all i, then the form is quadratic in x_i. In other words, a quadratic form is a general expression which contains second order terms.

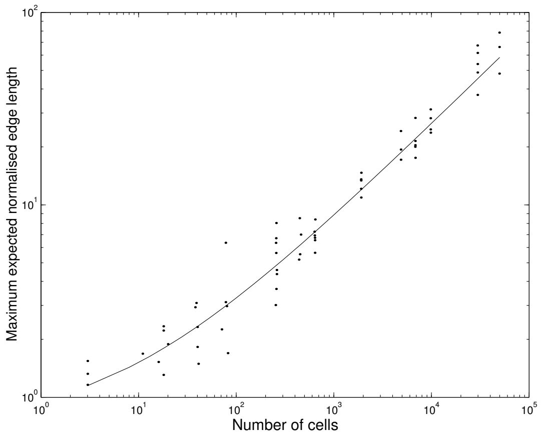
Voronoi statistics

In two dimensions the statistical descriptions are as follows.

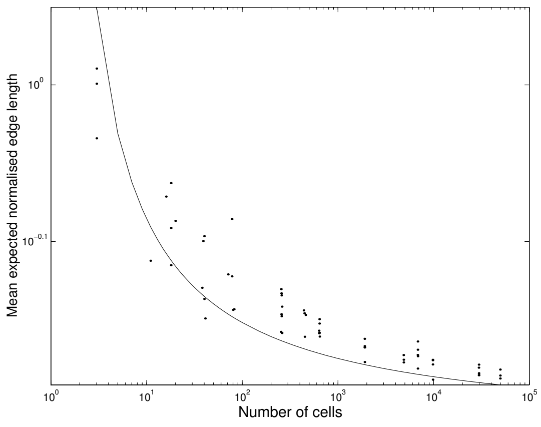
The curve is $y = 0.62/n^{0.13}$.: First the edge lengths are normalised by the edge length of the equivalent or characteristic square . A *equivalent square* is defined as the square figure whose area is equal to the area of the polygon in question, here a Voronoi polygon. Then find the average of the edge lengths in each cell.



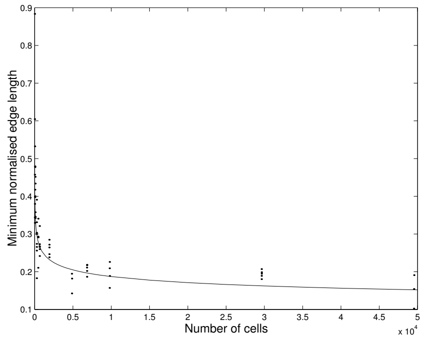
The curve is $y = \left|(n/15)^{1/2}\right| + 0.7$.: Of these cell-averaged normalised edge length obtained from simulations on various sizes of networks the minimum values are plotted in Figure 15, the maximum in Figure 16, and the expected value in Figure ℵ. Note that the last quantity is the average over the whole structure of all the averages obtained one from each cell.



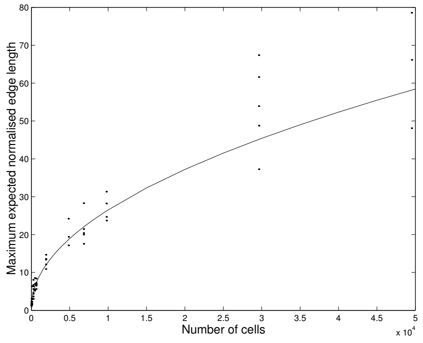
The curve is $y = 10^{0.2/\log x} - 0.4$.: To summarise, as the networks gets larger its minimum, maximum, and mean of the edge lengths when compared with the characteristic length approach constant values. The characteristic length is defined as $l = [\sum_{i=1}^{n} A_i/n]^{1/2}$, where A_i is the area of the ith polygon and n is the number of cells.



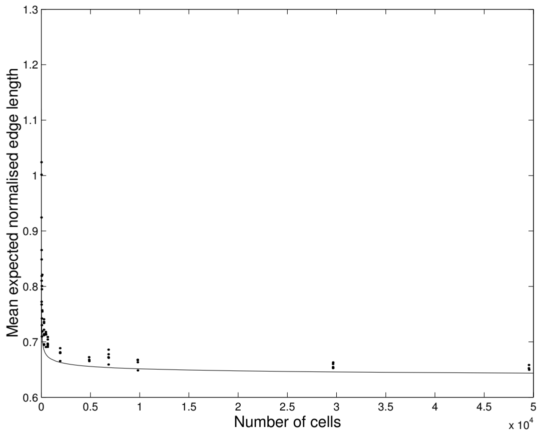
That the three values mentioned become constant may not seem obvious by the look of Figures 15 to 17 because the scale used there is a logarithmo-logarithmic scale, not a Euclidean one. These figures emphasise the smaller ranges of size. Figure 18 below on the other hand is plotted using a normal scale which enables one to see the asymptotic effect more clearly. Here the figures (a), (b), and (c) are respectively Figure 15, 16, and *manel*. Let the term *representative* stands for 'of the cell-average normalised', and *length* means 'edge length' in this context. Then the *minimum representative length* approaches 0.15 from Figure 18 (a), the *maximum representative length* is ever increasing, seemingly by a power law of approximately 0.5, while *mean representative length* approaches the value of 0.65. Properties of the Voronoi tessellation can be divided into individual and collective properties. With this in mind the term *length* above represents a property, *representative* means individual, and *minimum*, *maximum* and *mean* show the collective attributes.



(a)

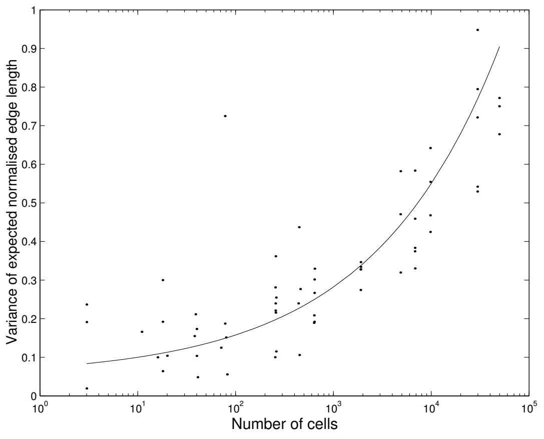


(b)

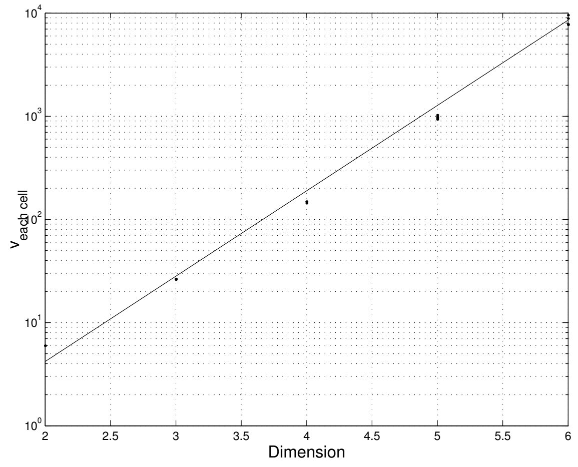


This $\sigma^2(\mathcal{N}_v^c(E(l_e)))$ increases very slowly with the increasing sizes of the networks. The curve shown has the equation

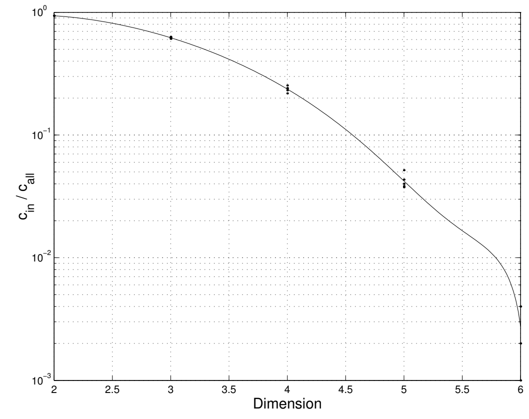
$$y = \left| \left[x/(8 \times 10^4) \right]^{1/3} \right| + 0.05$$



The number of vertices per cell increases dramatically as one goes up the dimension ladder. The programme used contains the essential part of the code which produces Figure 20. The straight line shown is $0.093(4+e)^n$. Notice the trend towards a greater rate of increase at dimensions higher than the maximum six shown. Only Voronoi cells which lies within the original domain and the vertices of these are considered. The same programme also gives Figure 21.



The expanse and the hypervolume of the Voronoi tessellation increases at an enormous rate, which results in the number of cells totally bounded within the original domain decreasing rapidly in Figure 21 as the dimension goes up. The effect at close to the zero ratio is emphasised by using the log scale for the y-axis. Also with the logarithmic scale the polynomial estimation curves that give negative values of the ratio is automatically excluded.

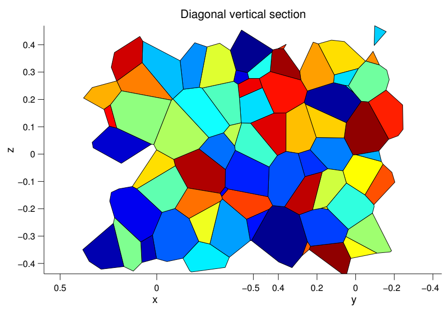


One can fit the polynomial $p(x) = p_1x^n + p_2x^{n-1} + \ldots + p_nx + p_{n+1}$ to the data with a least square algorithm. If $\mathbf{x}$ is the vector containing the data, then the the $(n+1)$ coefficients of the estimated polynomial can be found from $\hat{\mathbf{x}} = (\mathbf{x} - \mathrm{E}(\mathbf{x}))/\sigma(\mathbf{x})$. For data containing independent normal errors with a constant variance, the error bounds contain at least half of the predictions. The curve shown in Figure 21 is $p(x) = -0.048x^4 + 0.064x^3 - 0.203x^2 - 0.459x + 0.263$, the average value $\mathrm{E}(\mathbf{x})$ is 3.917, and the standard deviation $\sigma(\mathbf{x})$ is 1.412. The structure of the polynomial fit can be described using the Cholesky factor of the Vandermonde matrix

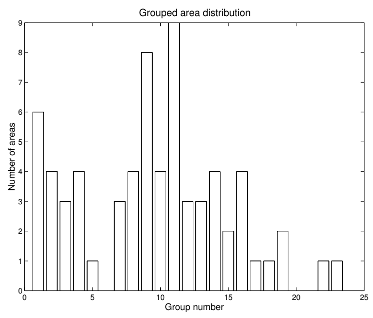
$$R = \begin{bmatrix} -12.19 & -1.56 & -6.08 & -0.47 & -3.17 \\ 0 & 8.46 & -0.45 & 4.47 & -0.44 \\ 0 & 0 & 1.21 & 0.33 & 2.93 \\ 0 & 0 & 0 & -1.63 & 0.30 \\ 0 & 0 & 0 & 0 & 2.25 \end{bmatrix},$$

the degree of freedom which is 19, and the norm of the residuals which is 0.034 in this case.

Voronoi section



(a)



(b)

Figure 22 *(a) Section by the plane $x - y + \epsilon z = \epsilon, \epsilon = 10^{-4}$. (b) Grouped distribution of area, the number of groups is approximately one-third the number of regions.*

	$\mathcal{V}_c$	$\aleph_c$	A_c^{fr}	V_c^{fr}	n_c^e	$E(n_{f,c}^e)$	$\sigma(n_{f,c}^e)$
min	8	6	2.2191×10^{-4}	9.9722×10^{-5}	12	4	0.6030
max	46	25	4.7015×10^{-3}	0.14364	69	5.52	2.5690
μ	26.338	15.169	1.8975×10^{-3}	1.8975×10^{-3}	39.507	5.1712	1.5586
σ^2	40.608	10.152	5.2144×10^{-7}	4.2313×10^{-5}	91.368	0.035983	0.1186
σ	6.3725	3.1862	7.2210×10^{-4}	6.5048×10^{-3}	9.5587	0.18969	0.3444
μ_g	25.543	14.829	1.7572×10^{-3}	1.1575×10^{-3}	38.315	5.1676	1.5169
μ_h	24.703	14.478	1.6026×10^{-3}	8.5548×10^{-4}	37.054	5.1638	1.4700
med	26	15	1.8125×10^{-3}	1.1660×10^{-3}	39	5.2	1.5706
mad	5.0068	2.5034	5.6975×10^{-4}	1.4250×10^{-3}	7.5103	0.14465	0.2715
$\mathcal{M}^2$	40.531	10.133	5.2045×10^{-7}	4.2233×10^{-5}	91.195	0.035915	0.1184
$\mathcal{M}^3$	72	9	2.5644×10^{-10}	5.4435×10^{-6}	243	-8.3404×10^{-3}	-0.0060
$\mathcal{M}^4$	5088.9	318.06	1.0467×10^{-12}	7.6653×10^{-7}	25763	7.9566×10^{-3}	0.041906
$\mathcal{K}$	3.0978	3.0978	3.8644	429.77	3.0978	6.1689	2.9916

Table 2 *Simulation uses rbox (1000 random points, seed 234985) and qhull (option v and o); $d = 3$, $n^c = 527$, $n^v = 6357$, CPU time 6,466.99 sec for the counting of statistics, 270.56 sec for finding area of the faces, 4.47 sec for calculating cell volume and 2.8 sec for finding the number of edges.*

Number of vertices and edges

It has been observed from the simulations that in three dimensions cells always have vertices in even numbers and edges odd ones. This can be explained by the following theorems.

Theorem 3. (*cf* Miles, 1972) In a simple three dimensional Voronoi tessellation, $3n_c^v = 2n_c^e$.

Proof.: Pick any Voronoi cell of the tessellation. Suppose that it has n^v vertices. Add up the number of edges connected to all vertices. Because every cell is a simple polyhedron, there are exactly three edges connected to each vertex. The number of edges thus counted is therefore $3n^v$. But each of the edges is connected to two vertices, so we have counted every one of them twice. Therefore,

$$2n^e = 3n^v.$$

This is the case for any cell, hence the theorem is proved. □

This theorem gives rise the following two theorems.

Theorem 4. The number of vertices of any cell within a simple three dimensional Voronoi tessellation is an even positive integer.

Proof.: Observe that the term $2n_c^e$ in the theorem above is divisible by 2. This term is equal to $3n_c^v$, therefore the latter is also divisible by 2. Since 2 can not divide into 3, the only term left, n_c^v, must be divisible by 2 and hence even number. □

Theorem 5. The number of edges of any cell within a simple three dimensional Voronoi tessellation is a positive integer divisible by three.

Proof.: With the same line of reasoning as above, observe that $3n_c^v$ is divisible by 3. Therefore $2n_c^e$, and hence n_c^e, is also divisible by 3. □

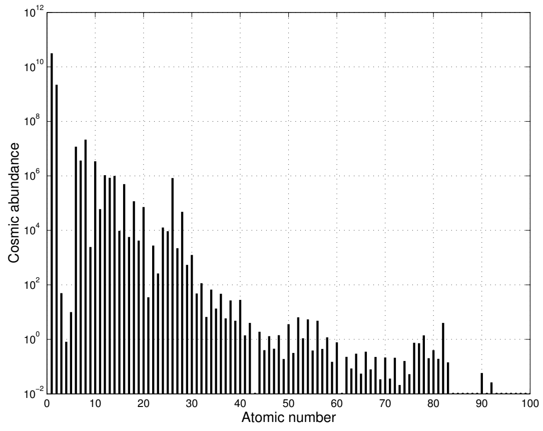
Another proof for both the above theorems is the following.

Proof.: For an equality to hold, both sides must have the same factors. By supposing an unknown common factor i and by cross-multiplying the coefficients on both sides, one obtains $2 \cdot (3 \cdot i) = 3 \cdot (2 \cdot i)$, where i is a positive integer. Therefore $n_c^e = 3i$ and $n_c^v = 2i$. In other words, n_c^e is divisible by three and n_c^v is even. □

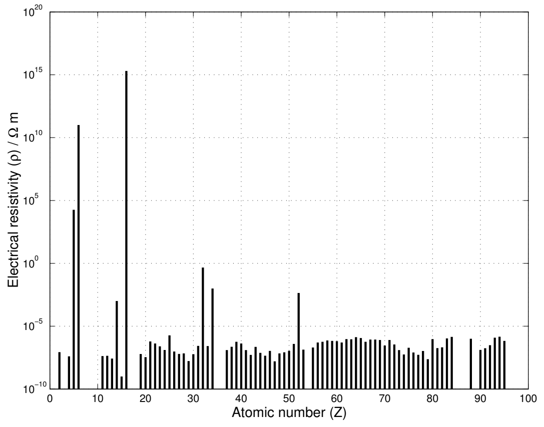
The theorems above assume that every cells are simple. This can not be the case in real situation where edges have dimensions and rather represent tubes than one-dimensional lines. Such case is similar to the so-called *degenerative* case in a computational model of Voronoi tessellation where there exist vertices the number of edges connected to each of which exceeds four. Even in the degenerative case, one would perhaps still expect a tendency for n_c^v to be an even number and for n_c^e to be divisible by three to hold.

In nature there are things which have a tendency towards even numbers. The following graph shows the the abundance in cosmic materials from the compilation by Cameron (1973). The year of publication of this paper is often misquoted as 1970, due to misprints in a footnote on the first page of the paper. Even the legendary Fred Hoyle has consistently practised this mistake and let it go uncorrected throughout his career. From what I have come across without an effort of searching, in two of his books and at least one of his papers, spanning the period of roughly forty years in total (*cf* Hoyle, 1977). Out of a sample of 278 of those papers which cite this work, this misprint resulted in 80% of the total number of errors in the year cited, which in turn amounts to 1.8% of the number of samples.

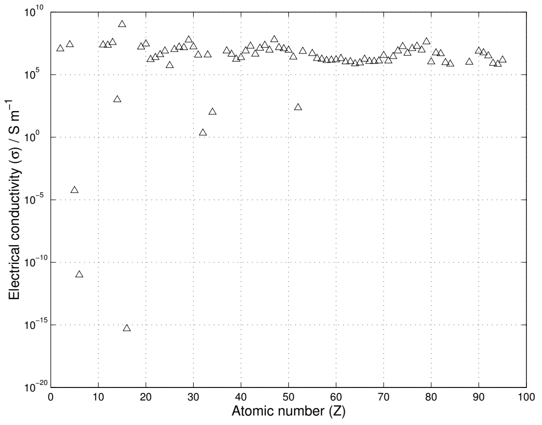
Figure 23 shows *The abundances of the chemical elements* in the universe. They are assumed to be the same as those found in the primitive solar nebula, which have been deduced from data on abundances found in chondritic meteorites and those found in the Sun. All abundances are relative to that of Si which is taken to be 10^6. Missing bars appear where the atomic numbers are unstable. Except for the atomic number 1 of Hydrogen, which is the most universal element, all other elements with even atomic numbers are locally more abundant than those with near-by odd atomic numbers.



Of interest are also the electrical resistivity and conductivity of solid matters. The conductivity σ is by definition the reciprocal of the resistivity ρ. Some of the solids, particularly boron, carbon, silicon, sulphur, germanium, selenium and tellurium, have a distinctively higher resistivity than the majority. Interestingly all of these, with only one exception of boron whose atomic number is five, are of an even atomic number, which respectively from carbon are 6, 14, 16, 32, 34, and 52. This can be seen in Figure 24 the data of which are taken from Podesta (2002).



(a)



(b)

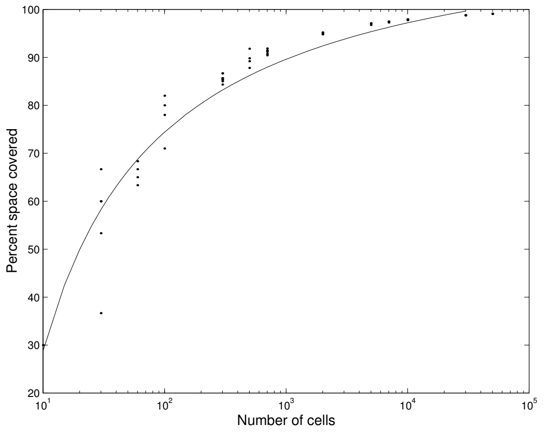
Figure 24 *The electrical resistivity, (a), and conductivity, (b), of elements which are solid at the room temperature.*

	n_c^e	$\mathcal{N}_v^c(E(l_c^e))$	$\mathcal{N}_v^c(\wp_c)$	$\mathcal{N}_v^c(A_c)$
min	3	0.20716	0.2749	0.058428
max	11	8.1072	10.134	22.307
μ	5.8973	0.70626	1.0089	1
σ^2	1.8955	0.18607	0.2942	1.4379
σ	1.3768	0.43136	0.54241	1.1991
μ_g	5.742	0.66036	0.94795	0.79225
μ_h	5.5904	0.62941	0.89999	0.61799
med	6	0.65709	0.96614	0.84267
mad	1.0736	0.17995	0.24357	0.49216
$\mathcal{M}^2$	1.8912	0.18565	0.29355	1.4347
$\mathcal{M}^3$	1.6194	0.96371	1.7931	22.703
$\mathcal{M}^4$	12.438	6.8376	15.785	468.03
$\mathcal{K}$	3.4775	198.38	183.18	227.39

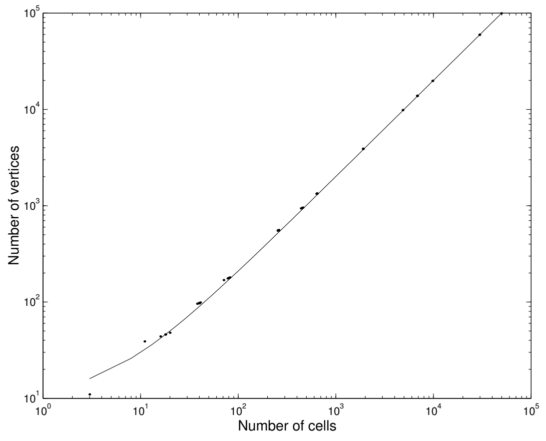
Table 3 *Neighbour statistics. Here for the normalisation purpose,* $l^e_{\text{basis}} = 0.046662$, $\wp_{\text{basis}} = 0.18665$, $A_{\text{basis}} = 0.0021773$. *Simulation uses voronoin command in Matlab;* $d = 2$, $n^c = 448$, $n^v = 946$, *CPU time 1.19 seconds.*

Next simulation was done with $d = 2$, $n^c = 3$ to 49551.

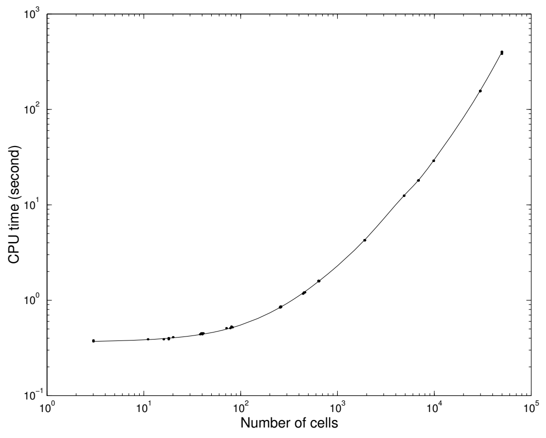
Figure 25 shows percentage of space covered by a Voronoi structure. The number of cells is the total number of cells generated. The percent space covered is the volume of the structure after boundary cells, that is cells which extrude the unit volume boundary, have been excluded. The equation of the reference curve is $y = -210/\log x + 120$.



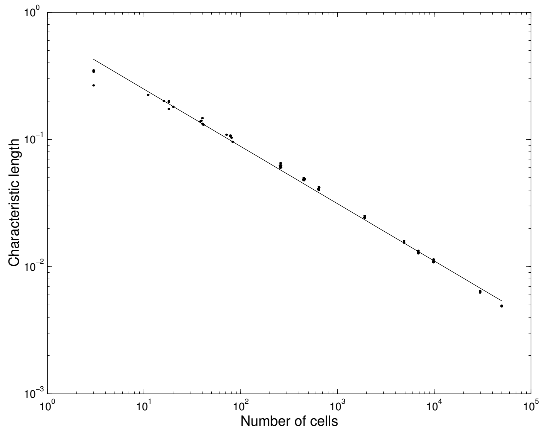
The reference line in Figure 26 is the linear equation $y = 2x + 10$. Boundary cells have been excluded.



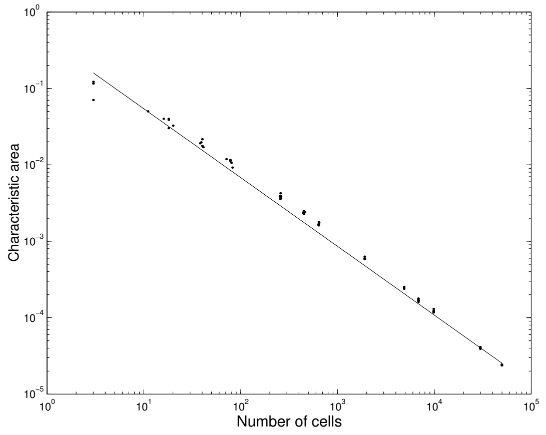
The curve in Figure 27 is the result of curve fitting by cubic spline interpolation.



The characteristic length is the length of the side of the cubic structure having the same number of cells and the same total volume as the Voronoi structure. The characteristic lengths in Figure 28 are shown as dots. The reference line is $y = 0.7/x^{0.45}$.

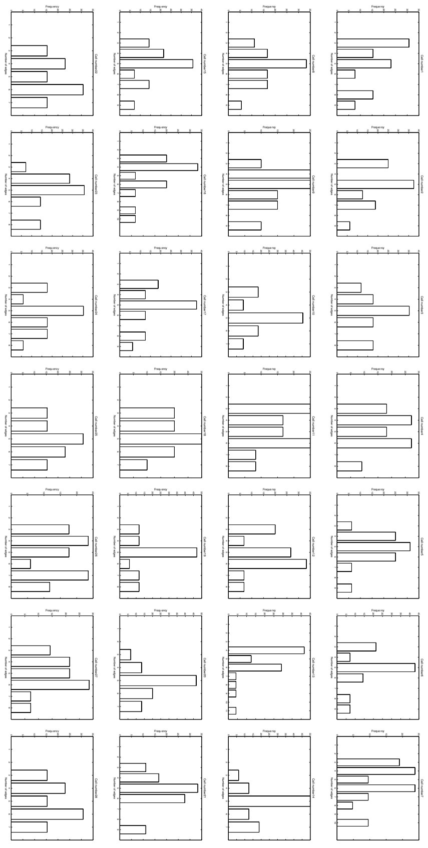


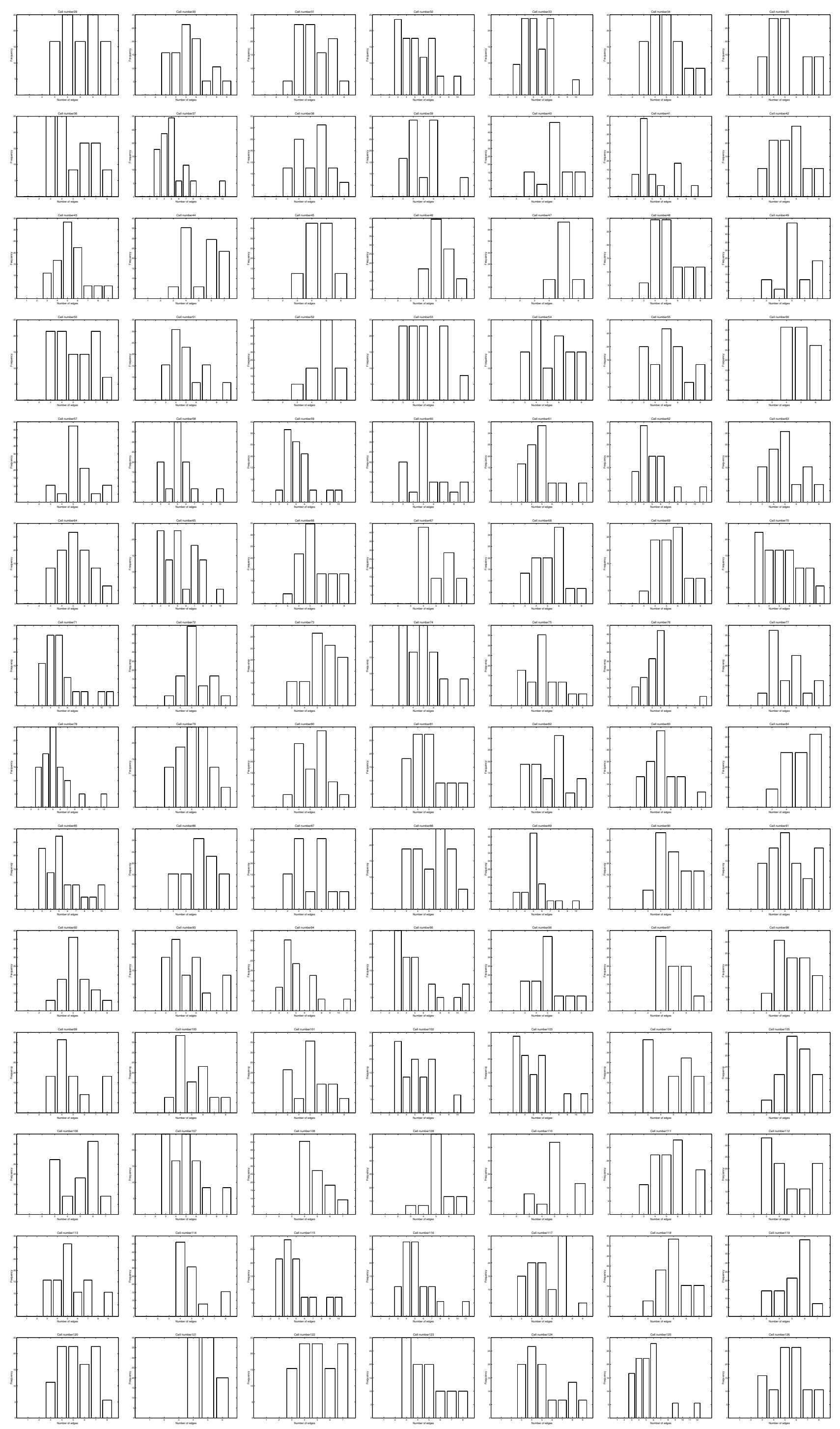
The characteristic area is the area of each square in the assembly the total volume and the number of cells of which are the same as those of the Voronoi graph. The reference curve shown in Figure 29 is $y = 0.43/x^{0.9}$.

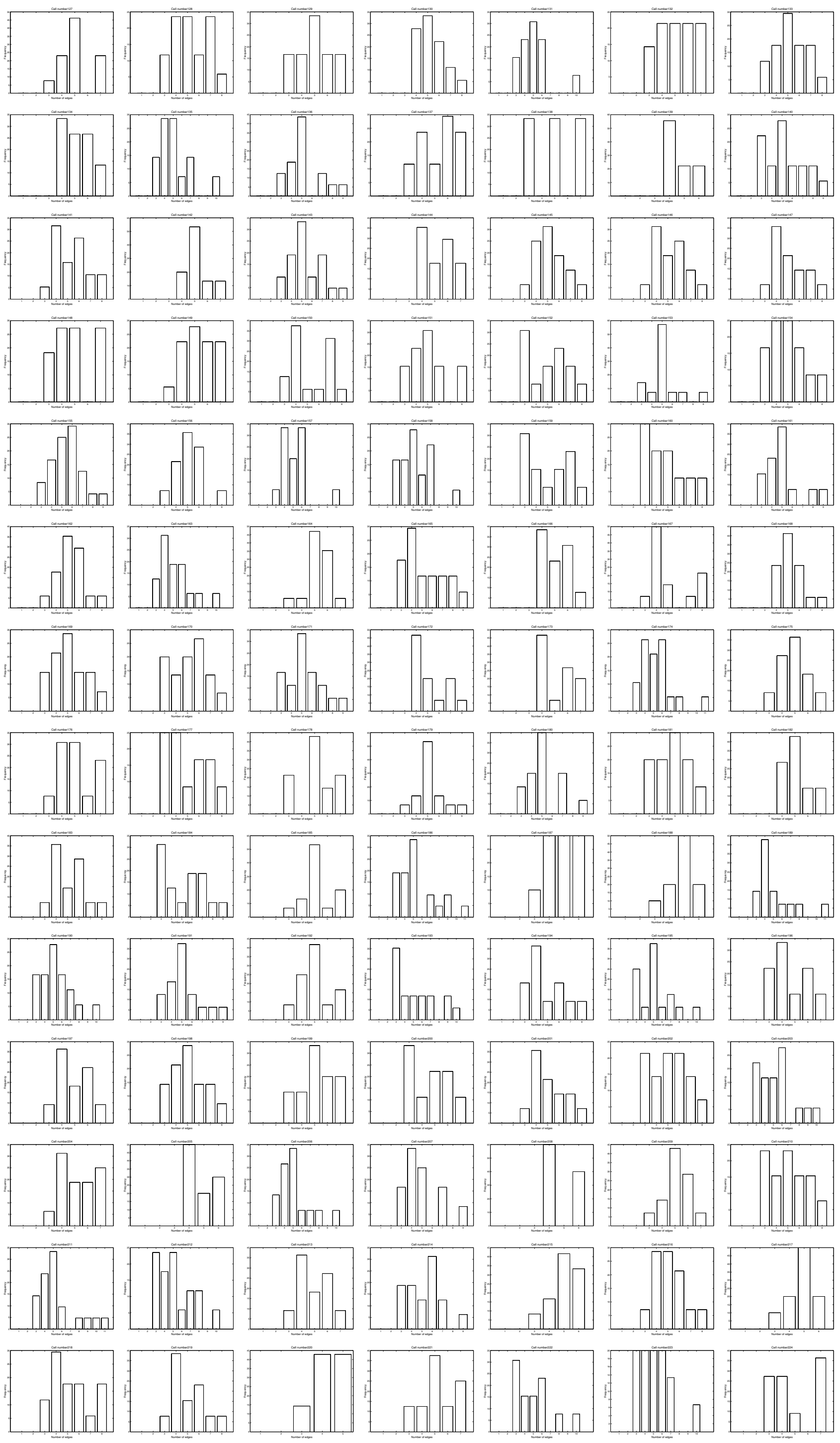


	n_c^v	n_f^v	$\mathcal{N}_v^c(\mathcal{P}_c^f)$	$\mathcal{N}_v^c(A_f)$	$\mathcal{N}_v^c(A_c)$	$\mathcal{N}_v^c(V_c)$	n_c^f
min	8	3	0.001414	4.6179×10^{-7}	0.11908	0.21464	6
max	42	10	0.84885	0.82919	0.50425	12.719	23
μ	24.423	5.1114	0.32876	0.12535	0.30315	1.0000	14.211
σ^2	44.105	2.1650	0.030844	0.015846	0.008511	2.5764	11.026
μ_g	23.504	4.9065	0.25897	0.051216	0.28815	0.70118	13.825
μ_h	22.506	4.7086	0.12620	2.50×10^{-4}	0.27198	0.58794	13.423
med	22	5	0.33219	0.084165	0.30368	0.64077	13
mad	5.4338	1.1655	0.14499	0.099433	0.075445	0.67950	2.7169
$\mathcal{M}^2$	43.483	2.1619	0.030800	0.015823	0.008391	2.5401	10.871
$\mathcal{M}^3$	112.13	1.8902	5.92×10^{-4}	0.002805	4.8×10^{-5}	24.169	14.016
$\mathcal{K}$	3.1096	3.0234	2.3090	5.2926	2.3015	42.238	3.1096

Table 4 *From rbox (200 random points, seed 34565473) and qhull (option v and o);* $d = 3, n^c = 71.$







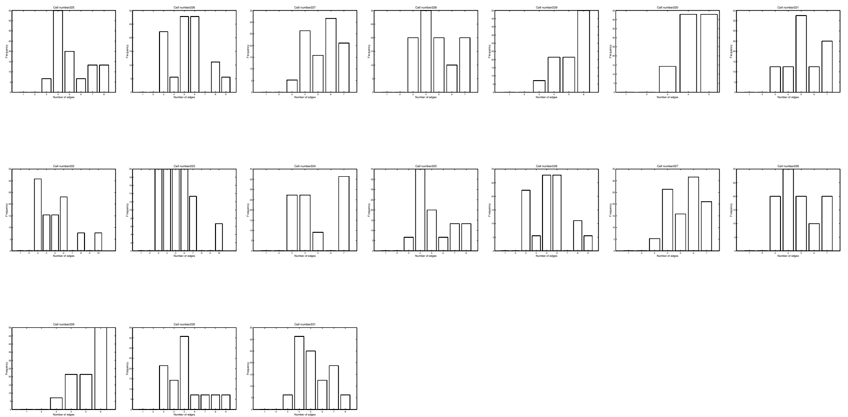


Figure 30 *Distribution of the number of edges per face. Each picture is an individual cell. The distribution shows the relative abundance or the number of faces (the vertical axes) having the number of edges as shown by the horizontal axes. The horizontal axis scales are positive integers starting from zero at the origin. Simulation uses rbox (500 random points, seed 893280) and qhull (option v and o); $d = 3$, $n^c = 231$, $n^v = 3{,}107$, CPU time 81.1 sec for finding face area, 2.22 sec for cell volume, 49.59 sec for counting edges and 0.03 sec for finding the number of edges and faces.*

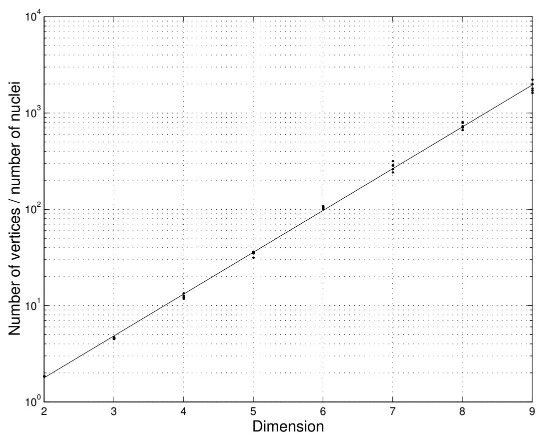
The minimum number of edges for each face is three. This number is the same as the number of vertices of that face. From these figures most of the cells have at least one face with three edges. There are only 19 cells (8.23 per cent) which does not have any three-edged face, and all of them have some four-edged faces. Therefore there is no cell with five as the minimum number of edges per face. The maximum number of edges per face is less clear-cut. There are twelve cells (5.19 per cent) with 11 as the maximum number of edges per face and two (0.87 per cent) with 12.

	n_c^v	$\aleph_c$	A_c^{fr}	V_c^{fr}	α	n_c^e	$E(n_{f,c}^e)$	$\sigma(n_{f,c}^e)$
min	10	7	1.0989×10^{-3}	2.5790×10^{-4}	0.0406	15	4.2857	0.6030
max	44	24	1.0956×10^{-2}	0.24840	4.2505	66	5.5000	2.7028
μ	25.974	14.987	4.3290×10^{-3}	4.3290×10^{-3}	1.2515	38.961	5.1601	1.5450
σ^2	40.852	10.213	3.3308×10^{-6}	2.9419×10^{-4}	0.3802	91.916	0.0372	0.1243
σ	6.3915	3.1958	1.8250×10^{-3}	1.7152×10^{-2}	0.6166	9.5873	0.1928	0.3526
μ_g	25.166	14.642	3.9633×10^{-3}	1.8465×10^{-3}	1.0816	37.749	5.1564	1.5030
μ_h	24.323	14.288	3.6026×10^{-3}	1.2471×10^{-3}	0.8241	36.484	5.1526	1.4580
med	26	15	4.1467×10^{-3}	1.5715×10^{-3}	1.1915	39	5.2000	1.5315
mad	5.1532	2.5766	1.4064×10^{-3}	4.5864×10^{-3}	0.4720	7.7298	0.1505	0.2794
$\mathcal{M}^2$	40.675	10.169	3.3163×10^{-6}	2.9291×10^{-4}	0.3785	91.518	0.0370	0.1238
$\mathcal{M}^3$	59.377	7.4222	5.7258×10^{-9}	6.3817×10^{-5}	0.2143	2.0040×10^2	-0.0071	0.0058
$\mathcal{M}^4$	4.5398×10^3	2.8374×10^2	4.6684×10^{-11}	1.5395×10^{-5}	0.7390	2.2983×10^4	0.0063	0.0463
$\mathcal{K}$	2.7440	2.7440	4.2448	1.7943×10^2	5.1578	2.7440	4.5730	3.0230

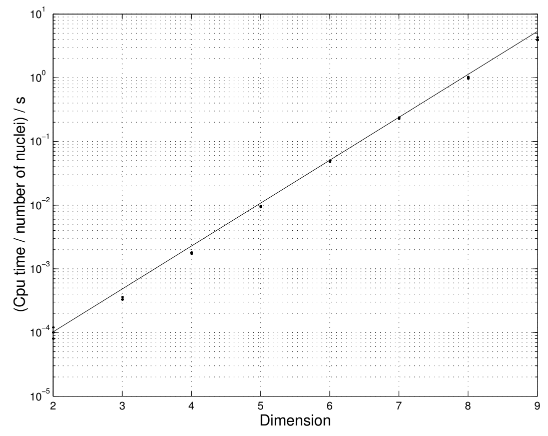
Figure 31 *Statistics of a 231 cells Voronoi structure.*

Because the Voronoi tessellation being studied is simple, $\aleph_c^v = \aleph_c^f = n_c^f$. In other words, any two cells having at least one vertex in common are neighbours to each other, and the number of neighbours around any cell in an infinite network is equal to the number of its faces.

The line shown in Figure 32 is $y = 0.2410e^n$, where n is the dimension of the network and is the horizontal axis; the coefficient of the exponential term is obtained by averaging over the averages in each dimension, each of which in turn comes from five batch runs.



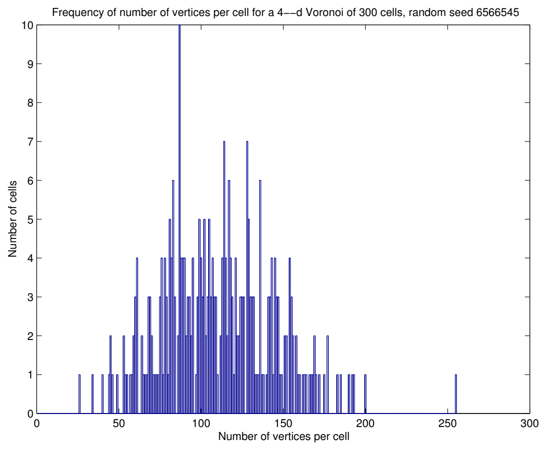
The line shown in Figure 33, found manually by trial and error, has the equation $y = 4.61 \times 10^{-6}(2 + e)^n$. In comparison, substituting the average cpu time for each dimension for y in the equation $y = Ae^n$ to obtain A and then find the average again over all dimensions results in $\bar{A} = 1.612 \times 10^{-4}$. The programme which produces both Figure 33 and 32 is given at the end.



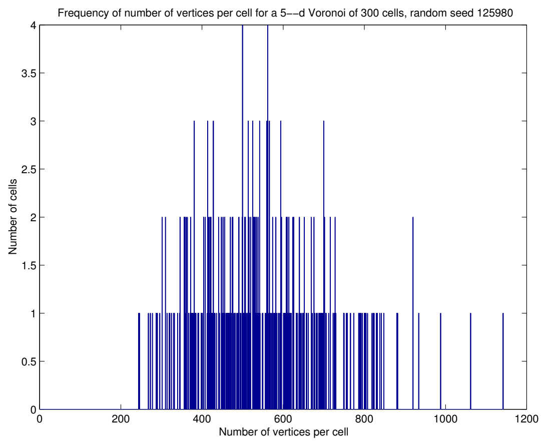
The number of vertices of higher dimensions is investigated briefly in the following Table 5 and Figure 34 the simulation of which was carried out by the programme listed at the end.

Dimension	*4*	*5*	*6*	*8*	*9*	*10*
N_c	300	300	300	30	30	30
N_v	6,577	27,150	118,534	5,465	10,467	17,442
n_c	132	104	14	2	3	0
$\min(n_c^v)$	26	244.00	1,020.0	952.002	2,353.0	4,287.0
$\max(n_c^v)$	255	1,142.0	5,852.0	2,638.0	5,534.0	9,528.0
$\bar{n}_c^v$	109.90	543.51	2,766.5	1,640.2	3,489.7	6,396.0
$(\sigma_{n_v}^2)_c$	1,155.9	22,974	6.0898×10^5	1.7964×10^5	7.0328×10^5	2.3635×10^6
$\sigma_c^{n_v}$	33.999	151.57	780.37	423.84	838.62	1.5374×10^3
$\mu_g(n_c^v)$	104.41	523.06	2,666.1	1,589.9	3,398.4	6,225.6
$\mu_h(n_c^v)$	98.389	502.94	2,571.4	1,542.3	3,313.9	6,065.2
$\text{med}(n_c^v)$	107.50	527.00	2,651.0	1,568.5	3,272.5	5,816.5
$\text{mad}(n_c^v)$	27.366	119.39	588.19	331.96	687.47	1,296.6
$m^2(n_c^v)$	1,152.1	22,897	6.0695×10^5	1.7365×10^5	6.7984×10^5	2.2847×10^6
$m^3(n_c^v)$	15,555	2.2871×10^6	5.1556×10^8	4.7965×10^7	3.9108×10^8	1.8988×10^9
$m^4(n_c^v)$	4.5760×10^6	1.8905×10^9	1.7964×10^{12}	8.5043×10^{10}	1.1967×10^{12}	1.1638×10^{13}
$\kappa(n_c^v)$	3.4476	3.6059	4.8764	2.8202	2.5892	2.2296

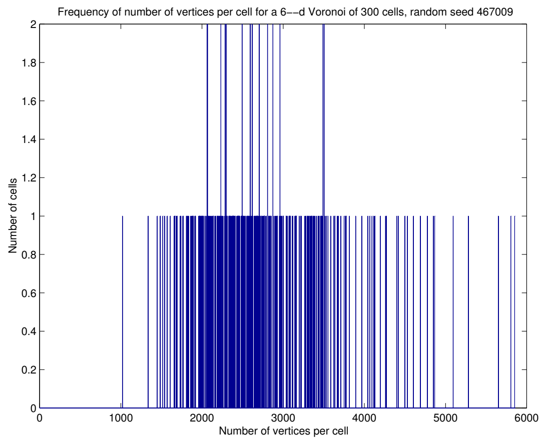
Table 5 *Statistics of the number of vertices in 4, 5, 6, 8, 9, and 10 dimensions*


Frequency of number of vertices per cell for a 4−−d Voronoi of 300 cells, random seed 6566545
Number of cells
Number of vertices per cell
0
1
2
3
4
5
6
7
8
9
10
0
50
100
150
200
250
300

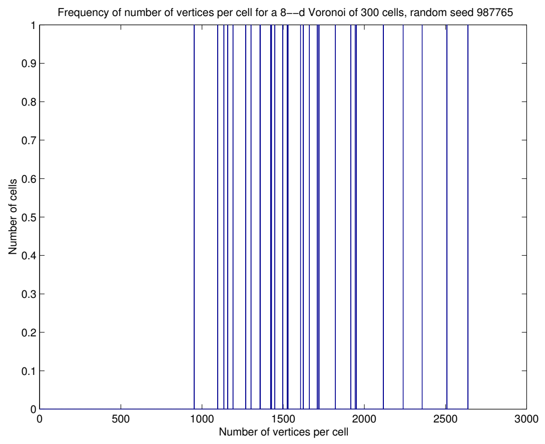
(a)


Frequency of number of vertices per cell for a 5−−d Voronoi of 300 cells, random seed 125980
Number of cells
Number of vertices per cell
0
0.5
1
1.5
2
2.5
3
3.5
4
0
200
400
600
800
1000
1200

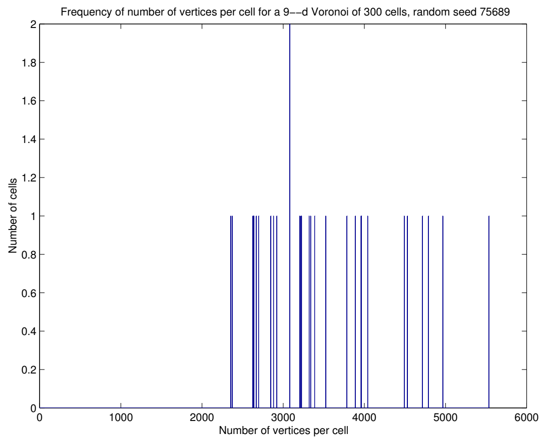
(b)


Frequency of number of vertices per cell for a 6−−d Voronoi of 300 cells, random seed 467009
Number of cells
Number of vertices per cell
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
0
1000
2000
3000
4000
5000
6000

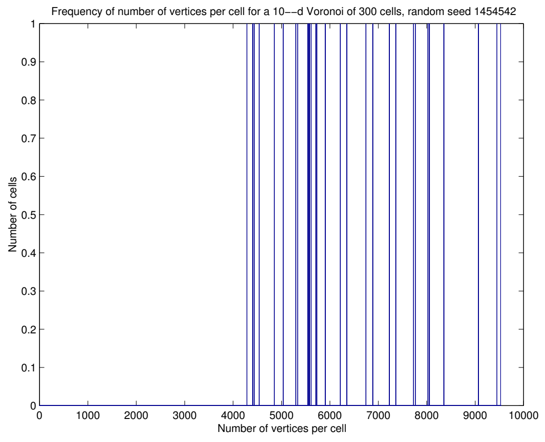
(c)


Frequency of number of vertices per cell for a 8−−d Voronoi of 300 cells, random seed 987765
Number of cells
Number of vertices per cell
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
500
1000
1500
2000
2500
3000

(d)


Frequency of number of vertices per cell for a 9−−d Voronoi of 300 cells, random seed 75689
Number of cells
Number of vertices per cell
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
1000
2000
3000
4000
5000
6000

(e)

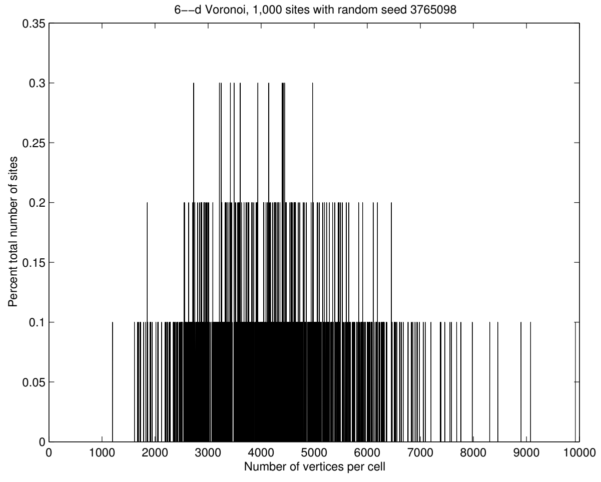

Frequency of number of vertices per cell for a 10−−d Voronoi of 300 cells, random seed 1454542
Number of cells
Number of vertices per cell
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000

(f)

Figure 34 *Distribution of vertices in* (a) *4,* (b) *5,* (c) *6, and* (d) *8,* (e) *9,* (f) *10 dimensions*

To obtain the number of vertices per cell, n_c^v, of a six-dimensional Voronoi structure of 1,000 cells I used a batch programme, for instance the one listed at the end. The Matlab macro that this programme refers to opens and reads from a file the number of vertices and then finds the statistical values, *viz.* $\min(n_c^v) = 1{,}198$; $\max(n_c^v) = 9{,}923$; $\bar{n}_c^v = 4{,}201.1$; $(\sigma_{n_v}^2)_c = 1.4069 \times 10^6$; $\sigma_c^{n_v} = 1186.1$; $(\bar{n}_g)_c^v = 4035.4$; $(\bar{n}_h)_c^v = 3866.8$; $\mathrm{med}(n_c^v) = 4122.5$; $\mathrm{mad}(n_c^v) = 931.81$; $m^2(n_c^v) = 1.4055 \times 10^6$; $m^3(n_c^v) = 1.0462 \times 10^9$; $m^4(n_c^v) = 7.7148 \times 10^{12}$; and $\kappa(n_c^v) = 3.9053$.

The following figure shows the frequency of the number of vertices.



There are three cells with 2729, 3213, 3246, 3421, 3490, 3606, 3938, 4143, 4398, 4417, 4442 and 4970 vertices. There are two having 1854, 2549, 2557, 2634, 2712, 2720, 2722, 2751, 2804, 2843, 2869, 2878, 2882, 2920, 2931, 2957, 2973, 2996, 3013, 3090, 3260, 3322, 3329, 3343, 3366, 3393, 3418, 3424, 3446, 3513, 3520, 3560, 3577, 3583, 3585, 3610, 3631, 3677, 3714, 3732, 3753, 3767, 3770, 3819, 3835, 3861, 3907, 3919, 3924, 3937, 3939, 4045, 4053, 4091, 4117, 4122, 4123, 4163, 4178, 4219, 4251, 4261, 4269, 4272, 4298, 4317, 4327, 4355, 4461, 4470, 4473, 4474, 4542, 4563, 4572, 4576, 4586, 4618, 4624, 4629, 4640, 4646, 4648, 4700, 4729, 4792, 4810, 4851, 4942, 4975, 4984, 4985, 5058, 5064, 5097, 5161, 5195, 5235, 5287, 5347, 5387, 5455, 5467, 5492, 5529, 5606, 5650, 5838, 5913, 6112, 6193 and 6455 vertices. And there is only one cell with each of the following numbers of vertices, 1198, 1612, 1672, 1686, 1688, 1717, 1727, 1787, 1827, 1864, 1906, 1915, 1921, 1947, 2016, 2052, 2056, 2060, 2123, 2190, 2200, 2223, 2231, 2236, 2267, 2280, 2281, 2286, 2289, 2344, 2353, 2356, 2371, 2372, 2385, 2408, 2423, 2426, 2429, 2455, 2475, 2487, 2499, 2500, 2503, 2504, 2508, 2513, 2542, 2547, 2551, 2560, 2561, 2565, 2575, 2578, 2580, 2585, 2586, 2596, 2617, 2619, 2622, 2650, 2654, 2658, 2664, 2669, 2673, 2686, 2692, 2694, 2704, 2708, 2716, 2718, 2723, 2732, 2733, 2742, 2743, 2755, 2771, 2779, 2781, 2783, 2791, 2800, 2807, 2808, 2811, 2820, 2835, 2837, 2853, 2858, 2864, 2867, 2870, 2873, 2875, 2880, 2884, 2887, 2893, 2894, 2895, 2902, 2907, 2910, 2914, 2916, 2925, 2928, 2934, 2949, 2955, 2956, 2960, 2967, 2974, 2978, 2983, 2985, 2989, 2993, 3007, 3012, 3014, 3027, 3051, 3074, 3102, 3107, 3114, 3116, 3119, 3129, 3130, 3136, 3137, 3141, 3148, 3152, 3164, 3173, 3176, 3180, 3182, 3186, 3188, 3190, 3192, 3199, 3202, 3204, 3206, 3208, 3219, 3221, 3229, 3231, 3237, 3239, 3244, 3245, 3255, 3256, 3259, 3267, 3270, 3271, 3274, 3275, 3285, 3295, 3296, 3305, 3307, 3312, 3316, 3317, 3318, 3319, 3333, 3337, 3344, 3351, 3352, 3354, 3357, 3360, 3370, 3373, 3374, 3378, 3384, 3390, 3394, 3396, 3402, 3403, 3408, 3427, 3428, 3436, 3440, 3443, 3444, 3452, 3453, 3454, 3486, 3487, 3492, 3499, 3502, 3505, 3506, 3509, 3519, 3521, 3522, 3523, 3536, 3541, 3544, 3545, 3547, 3549, 3551, 3555, 3556, 3573, 3575, 3578, 3581, 3582, 3587, 3589, 3594, 3595, 3599, 3604, 3609, 3611, 3620, 3628, 3634, 3635, 3636, 3639, 3647, 3652, 3656, 3658, 3663, 3666, 3668, 3680, 3685, 3686, 3687, 3689, 3691, 3694, 3695, 3696, 3706, 3709, 3710, 3713, 3716, 3717, 3727, 3728, 3735, 3739, 3740, 3742, 3743, 3744, 3751, 3764, 3772, 3775, 3776, 3778, 3782, 3787, 3789, 3792, 3794, 3798, 3803, 3804, 3806, 3808, 3810, 3815, 3818, 3821, 3830, 3834, 3837, 3838, 3842, 3847, 3849, 3850, 3863, 3867, 3868, 3883, 3884, 3889, 3890, 3899, 3900, 3902, 3903, 3908, 3912, 3914, 3918, 3920, 3925, 3929, 3942, 3943, 3944, 3947, 3949, 3956, 3957, 3960, 3962, 3963, 3978, 3980, 3981, 3986, 3988, 3996, 4008, 4016, 4017, 4019, 4027, 4030, 4033, 4035, 4038, 4039, 4044, 4049, 4057, 4072, 4073, 4075, 4077, 4082, 4086, 4092, 4106, 4111, 4112, 4124, 4126, 4128, 4134, 4140, 4145, 4146, 4147, 4148, 4150, 4153, 4157, 4162, 4176, 4177, 4179, 4188, 4189, 4190, 4193, 4195, 4203, 4206, 4212, 4214, 4217, 4218, 4225, 4227, 4232, 4238, 4239, 4244, 4245, 4246, 4247, 4257, 4262, 4271, 4273, 4279, 4283, 4288, 4291, 4293, 4300, 4308, 4313, 4316, 4320, 4322, 4324, 4326, 4331, 4334, 4335, 4337, 4339, 4342, 4346, 4349, 4350, 4351, 4352, 4354, 4357, 4361, 4376, 4377, 4386, 4388, 4395, 4399, 4405, 4410, 4411, 4413, 4425, 4426, 4428, 4447, 4449, 4451, 4453, 4459, 4471, 4475, 4480, 4488, 4493, 4496, 4501, 4504, 4506, 4512, 4515, 4518, 4522, 4523, 4532, 4548, 4552, 4556, 4561, 4574, 4580, 4585, 4587, 4595, 4598, 4602, 4605, 4610, 4614, 4619, 4621, 4626, 4628, 4632, 4634, 4639, 4649, 4651, 4657, 4663, 4665, 4674, 4681, 4684, 4697, 4702, 4703, 4705, 4710, 4711, 4714, 4725, 4728, 4736, 4738, 4740, 4751, 4754, 4755, 4761, 4765, 4770, 4779, 4781, 4814, 4816, 4826, 4829, 4831, 4844, 4850, 4873, 4877, 4880, 4885, 4890, 4892, 4896, 4899, 4900, 4902, 4903, 4905, 4908, 4911, 4914, 4915, 4919, 4924, 4933, 4937, 4944, 4955, 4960, 4967, 4969, 4974, 4992, 4994, 4997, 5000, 5002, 5003, 5007, 5019, 5021, 5028, 5029, 5035, 5039, 5050, 5074, 5080, 5083, 5092, 5093, 5104, 5108, 5110, 5117, 5119, 5120, 5131, 5138, 5156, 5158, 5166, 5167, 5176, 5185, 5192, 5194, 5200, 5203, 5219, 5225, 5226, 5233, 5236, 5239, 5251, 5259, 5265, 5270, 5271, 5274, 5285, 5288, 5290, 5313, 5316, 5318, 5325, 5328, 5336, 5337, 5346, 5364, 5389, 5405, 5408, 5415, 5432, 5439, 5440, 5448, 5457, 5460, 5463, 5464, 5466, 5468, 5470, 5474, 5482, 5484, 5530, 5549, 5567, 5589, 5591, 5598, 5612, 5627, 5632, 5664, 5676, 5680, 5718, 5719, 5720, 5724, 5738, 5744, 5747, 5748, 5749, 5752, 5764, 5802, 5806, 5824, 5841, 5859, 5880, 5885, 5892, 5896, 5899, 5927, 5928, 5947, 5956, 5957, 5966, 5998, 6006, 6014, 6017, 6031, 6035, 6038, 6040, 6054, 6055, 6075, 6081, 6084, 6089, 6092, 6113, 6145, 6166, 6169, 6180, 6192, 6195, 6221, 6237, 6265, 6272, 6273, 6286, 6303, 6305, 6310, 6313, 6320, 6356, 6367, 6456, 6469, 6510, 6515, 6520, 6521, 6524, 6541, 6561, 6609, 6615, 6642, 6677, 6767, 6771, 6831, 6852, 6883, 6918, 6945, 6985, 7056, 7093, 7201, 7379, 7387, 7461, 7560, 7588, 7685, 7764, 7765, 7979, 8308, 8460, 8899, 9079 and 9923. But these results are not very useful since border cells are present.

§ **Result on 700 cells**

```
Box size: 10
No compression
Number of cells: 700
Number of vertices: 4380
Number of cells in frame: 341
Time for counting stats: 2191.39 seconds
Number of faces connected to the first vertice at infinity: 181
Time for finding cell volumes: 3.1500 seconds
```

n_c	n_v	$n_{f,1^{st}v}$	$n_{c_{in}}$	$t_{CPU,stat.}$	$t_{CPU,A}$	$t_{CPU,V}$	
700	4,380	181	341	2191.39 sec.	137.13 sec.	3.15 sec.	
$\min n_{v_c}$	$\max n_{v_c}$	$\bar{n_{v_c}}$	$\sigma^2_{n_{v_c}}$	$\sigma_{n_{v_c}}$	$g, \bar{n}_{v_c}$	$h, \bar{n}_{v_c}$	n_{v_c}
10	46	25.106	46.784	6.8399	24.169	23.208	
$\widetilde{n_{v_c}}$	$\delta_\mu(n_{v_c})$	$M^2(n_{v_c})$	$M^3(n_{v_c})$	$M^4(n_{v_c})$	$\kappa_{n_v,c}$		
24	5.4023	46.717	1.4240×10^2	6.5877×10^3	3.0184		
$\min n_{v,c_{in}}$	$\max n_{v,c_{in}}$	$\bar{v_{c_{in}}}$	$\sigma^2_{v,c_{in}}$	$\sigma_{v,c_{in}}$	$\bar{V}_{,gc_{in}}$	$\bar{V}_{,hc_{in}}$	$n_{v,c_{in}}$
10	46	26.246	40.974	6.4011	25.464	24.668	
$\widetilde{n}_{v,c_{in}}$	$\delta_\mu(n_{v,c_{in}})$	$M^2(n_{v,c_{in}})$	$M^3(n_{v,c_{in}})$	$M^4(n_{v,c_{in}})$	$\kappa_{n_{v,c_{in}}}$		
26	5.1101	40.854	1.0338×10^2	4.9173×10^3	2.9461		
$\min n_{\aleph_v,c_{in}}$	$\max n_{\aleph_v,c_{in}}$	$\bar{n}_{\aleph_v,c_{in}}$	$\sigma^2_{n_{\aleph_v,c_{in}}}$	$\sigma_{n_{\aleph_v,c_{in}}}$	$\bar{n}_{g,\aleph_v,c_{in}}$	$\bar{n}_{h,\aleph_v,c_{in}}$	$n_{\aleph_v,c_{in}}$
8	26	16.123	10.244	3.2006	15.809	15.495	
$\widetilde{n}_{\aleph_v,c_{in}}$	$\delta_\mu(n_{\aleph_v,c_{in}})$	$M^2(n_{\aleph_v,c_{in}})$	$M^3(n_{\aleph_v,c_{in}})$	$M^4(n_{\aleph_v,c_{in}})$	$\kappa_{n_{\aleph_v,c_{in}}}$		
16	2.5550	10.214	12.922	3.0733×10^2	2.9461		
$\min n_{\aleph_e,c}$	$\max n_{\aleph_e,c}$	$\bar{n}_{\aleph_e,c}$	$\sigma^2_{n,\aleph_e,c}$	$\sigma_{n,\aleph_e,c}$	$\bar{n}_{g,\aleph_e,c}$	$\bar{n}_{h,\aleph_e,c}$	$n_{\aleph_e,c}$
8	27	15.669	12.439	3.5269	15.279	14.894	
$\widetilde{n}_{\aleph_e,c}$	$\delta_\mu(n_{\aleph_e,c})$	$M^2(n_{\aleph_e,c})$	$M^3(n_{\aleph_e,c})$	$M^4(n_{\aleph_e,c})$	$\kappa_{n,\aleph_e,c}$		
15	2.7792	12.422	22.761	4.8871×10^2	3.1673		
$\min n_{\aleph_e,c_{in}}$	$\max n_{\aleph_e,c_{in}}$	$\bar{n}_{\aleph_e,c_{in}}$	$\sigma^2_{n(\aleph_e),c_{in}}$	$\sigma_{n(\aleph_e),c_{in}}$	$\bar{n}_{g,\aleph_e,c_{in}}$	$\bar{n}_{h,\aleph_e,c_{in}}$	$n_{\aleph_e,c_{in}}$

8	26	16.123	10.244	3.2006	15.809	15.495	
$\widetilde{n}_{\aleph_e,c_{in}}$	$\delta_\mu(n_{\aleph_e,c_{in}})$	$M^2(n_{\aleph_e,c_{in}})$	$M^3(n_{\aleph_e,c_{in}})$	$M^4(n_{\aleph_e,c_{in}})$	$\kappa_{n(\aleph_e),c_{in}}$		
16	2.5550	10.214	12.922	3.0733×10^2	2.9461		
$\min n_{\aleph_f,c}$	$\max n_{\aleph_f,c}$	$\bar{n}_{\aleph_f,c}$	$\sigma^2_{n(\aleph_f),c}$	$\sigma_{n(\aleph_f),c}$	$\bar{n}_{g,\aleph_f,c}$	$\bar{n}_{h,\aleph_f,c}$	$n_{\aleph_f,c}$
8	27	15.629	12.274	3.5034	15.242	14.857	
$\widetilde{n}_{\aleph_f,c}$	$\delta_\mu(n_{\aleph_f,c})$	$M^2(n_{\aleph_f,c})$	$M^3(n_{\aleph_f,c})$	$M^4(n_{\aleph_f,c})$	$\kappa_{n(\aleph_f),c}$		
15	2.7635	12.256	21.051	4.6791×10^2	3.1149		
$\min n_{\aleph_f,c_{in}}$	$\max n_{\aleph_f,c_{in}}$	$\bar{n}_{\aleph_f,c_{in}}$	$\sigma^2_{n(\aleph_f),c_{in}}$	$\sigma_{n(\aleph_f),c_{in}}$	$\bar{n}_{g,\aleph_f,c_{in}}$	$\bar{n}_{h,\aleph_f,c_{in}}$	$n_{\aleph_f,c_{in}}$
8	26	16.123	10.244	3.2006	15.809	15.495	
$\widetilde{n}_{\aleph_f,c_{in}}$	$\delta_\mu(n_{\aleph_f,c_{in}})$	$M^2(n_{\aleph_f,c_{in}})$	$M^3(n_{\aleph_f,c_{in}})$	$M^4(n_{\aleph_f,c_{in}})$	$\kappa_{n(\aleph_f),c_{in}}$		
16	2.5550	10.214	12.922	3.0733×10^2	2.9461		
$\min A_c$	$\max A_c$	$\bar{A}_c$	$\sigma^2_{A_c}$	σ_{A_c}	$\bar{A}g,c$	$\bar{A}h,c$	A_c
0.59209	83.165	3.4086	2.0373	2.3046×10^2	5.0726	1.0502	
$\widetilde{A}_c$	$\delta_\mu(A_c)$	$M^2(A_c)$	$M^3(A_c)$	$M^4(A_c)$	κ_{A_c}		
1.1935×10^2	4.0384	2.8300	2.0530	1.3800	2.0572		
$\min A_{c_{in}}$	$\max A_{c_{in}}$	$\bar{A}_{c_{in}}$	$\sigma^2_{A_{c_{in}}}$	$\sigma_{A_{c_{in}}}$	$\bar{A}g,c_{in}$	$\bar{A}h,c_{in}$	$A_{c_{in}}$
0.59209	3.4086	2.0373	1.0502	4.0384	2.8300	2.0530	
$\widetilde{A}_{c_{in}}$	$\delta_\mu(A_{c_{in}})$	$M^2(A_{c_{in}})$	$M^3(A_{c_{in}})$	$M^4(A_{c_{in}})$	$\kappa_{A_{c_{in}}}$		
1.3800	2.0572	2.9782	3.9125	3.1031	1.0800		
$\min A_{fr,c}$	$\max A_{fr,c}$	$\bar{A}_{fr,c}$	$\sigma^2_{A_{fr,c}}$	$\sigma_{A_{fr,c}}$	$\bar{A}g,fr,c$	$\bar{A}h,fr,c$	$A_{fr,c}$
8.5015×10^{-8}	1.1941×10^{-5}	4.8942×10^{-7}	2.9253×10^{-7}	3.3091×10^{-5}	7.2835×10^{-7}	1.5079×10^{-7}	

$\widetilde{A}_{fr,c}$ 1.7137×10^{-5}	$\delta_\mu(A_{fr,c})$ 5.7986×10^{-7}	$M^2(A_{fr,c})$ 4.0635×10^{-7}	$M^3(A_{fr,c})$ 2.9478×10^{-7}	$M^4(A_{fr,c})$ 1.9816×10^{-7}	$\kappa_{A_{fr,c}}$ 2.9538×10^{-7}		
$\min A_{fr,c_{in}}$ 7.5700×10^{-4}	$\max A_{fr,c_{in}}$ 4.3579×10^{-3}	$\bar{A}_{fr,c_{in}}$ 2.6048×10^{-3}	$\sigma^2_{A_{fr,c_{in}}}$ 1.3427×10^{-3}	$\sigma_{A_{fr,c_{in}}}$ 5.1632×10^{-3}	$\bar{A}g,fr,c_{in}$ 3.6182×10^{-3}	$\bar{A}h,fr,c_{in}$ 2.6248×10^{-3}	$A_{fr,c_{in}}$
$\widetilde{A}_{fr,c_{in}}$ 1.7644×10^{-3}	$\delta_\mu(A_{fr,c_{in}})$ 2.6301×10^{-3}	$M^2(A_{fr,c_{in}})$ 3.8077×10^{-3}	$M^3(A_{fr,c_{in}})$ 5.0023×10^{-3}	$M^4(A_{fr,c_{in}})$ 3.9674×10^{-3}	$\kappa_{A_{fr,c_{in}}}$ 1.3808×10^{-3}		
$\min V_c$ 0.88511	$\max V_c$ 6.2052×10^{6}	$\bar{V}_c$ 2.7455×10^{4}	$\sigma^2_{V_c}$ 8.9303×10^{10}	σ_{V_c} 2.9884×10^{5}	$\bar{V}g,c$ 22.548	$\bar{V}h,c$ 6.3296	V_c
$\widetilde{V}_c$ 7.6212	$\delta_\mu(V_c)$ 5.1126×10^{4}	$M^2(V_c)$ 8.9176×10^{10}	$M^3(V_c)$ 4.3202×10^{17}	$M^4(V_c)$ 2.3781×10^{24}	κ_{V_c} 2.9904×10^{2}		
$\min V_{c_{in}}$ 0.88511	$\max V_{c_{in}}$ 1.8958×10^{3}	$\bar{V}_{c_{in}}$ 20.542	$\sigma^2_{V_{c_{in}}}$ 1.5264×10^{4}	$\sigma_{V_{c_{in}}}$ 1.2355×10^{2}	$\bar{V}g,c_{in}$ 5.9195	$\bar{V}h,c_{in}$ 4.4392	$V_{c_{in}}$
$\widetilde{V}_{c_{in}}$ 5.4355	$\delta_\mu(V_{c_{in}})$ 27.858	$M^2(V_{c_{in}})$ 1.5219×10^{4}	$M^3(V_{c_{in}})$ 2.2911×10^{7}	$M^4(V_{c_{in}})$ 3.9178×10^{10}	$\kappa_{V_{c_{in}}}$ 1.6915×10^{2}		
$\min V_{fr,c}$ 4.6056×10^{-8}	$\max V_{fr,c}$ 0.32288	$\bar{V}_{fr,c}$ 1.4286×10^{-3}	$\sigma^2_{V_{fr,c}}$ 2.4179×10^{-4}	$\sigma_{V_{fr,c}}$ 1.5550×10^{-2}	$\bar{V}g,fr,c$ 1.1733×10^{-6}	$\bar{V}h,fr,c$ 3.2935×10^{-7}	$V_{fr,c}$
$\widetilde{V}_{fr,c}$ 3.9656×10^{-7}	$\delta_\mu(V_{fr,c})$ 2.6603×10^{-3}	$M^2(V_{fr,c})$ 2.4145×10^{-4}	$M^3(V_{fr,c})$ 6.0865×10^{-5}	$M^4(V_{fr,c})$ 1.7433×10^{-5}	$\kappa_{V_{fr,c}}$ 2.9904×10^{2}		
$\min V_{fr,c_{in}}$ 1.2636×10^{-4}	$\max V_{fr,c_{in}}$ 0.27064	$\bar{V}_{fr,c_{in}}$ 2.9326×10^{-3}	$\sigma^2_{V_{fr,c_{in}}}$ 3.1108×10^{-4}	$\sigma_{V_{fr,c_{in}}}$ 1.7638×10^{-2}	$\bar{V}g,fr,c_{in}$ 8.4507×10^{-4}	$\bar{V}h,fr,c_{in}$ 6.3374×10^{-4}	$V_{fr,c_{in}}$
$\widetilde{V}_{fr,c_{in}}$ 7.7598×10^{-4}	$\delta_\mu(V_{fr,c_{in}})$ 3.9770×10^{-3}	$M^2(V_{fr,c_{in}})$ 3.1017×10^{-4}	$M^3(V_{fr,c_{in}})$ 6.6659×10^{-5}	$M^4(V_{fr,c_{in}})$ 1.6273×10^{-5}	$\kappa_{V_{fr,c_{in}}}$ 1.6915×10^{2}		

§ **AVS**

The first picture I created on AVS was a Trigonal Dipyramidal. The programme was the following .inp file.

```
5 2 0 0 0
1 0 0 1.2910
2 2 0 1.2910
3 1 1.7321 1.2910
4 1 0.5774 0
5 1 0.5774 2.5820
1 1 tet 1 2 3 4
2 1 tet 1 2 3 5
1 2
```

The connection of modules was ReadUCD to ExternalEdges to UViewer3D.

§ **Problems related to ISD and NQS**

Some of the staffs at our Information Systems Department are unprofessional in their jobs. Nicholas, for example, has once wiped out all about seven of my batch jobs on Network Queueing Systems. He emailed me on Monday 19th March that he had deleted one of my jobs because he thought it was doing nothing and consuming no CPU time. In his email he said it would not be able to reach its timeout limit in order to die out.

I knew what he had written could be true and little minded the fact that he killed my task before bother telling me about it first. Two minutes after having read his email I found out at the ISD department that the number was not one but all, about seven in total, of my jobs had been deleted. Zaiem agreed at that point that it was not a right thing for a system administrator to do to delete users' jobs without letting the owners know first.

What happened in this case was this. Two weeks ago one of my NQS jobs correctly produced an input file of larger than 300 MB in size before it died due to lack of resource. I suddenly realised then that another one or two of my other jobs, if allowed to go on further, would produce output files of larger than 1 GB in size each. I tried to kill them using `qdel`. I found that I could not do so, so I asked Zaiem about it. For two weeks we tried everything in vain. We could not delete either one of them. Meanwhile, I did the only thing possible for me to do in order to prevent larger output files from them. Not being able to kill the job, I deleted their input files instead.

It must have been for this reason that one of them went into a sleeping state as Nicholas said. But a sleeping programme could go on for a long time trying to read a nonexisting file without causing much trouble or even consuming much CPU time.

When Nicholas came on the phone that morning, however, he completely changed the story. Instead of saying that one of my jobs had been sleeping, it turned out to be that my jobs had *messed* up the whole NQS systems. He told me not to use NQS again after this.

It is one thing to say that a job sleeps; it is another to say that it disrupts. In fact I think they are as opposite to one another as *dull* and *vicious*.

I have just proved that two of my programmes which Nicholas killed actually completed very quickly when run interactively. I waited while the two ran in the foreground. One (f72.m) completed within 10 minutes while the other (f71.m) took less than one hour to run.

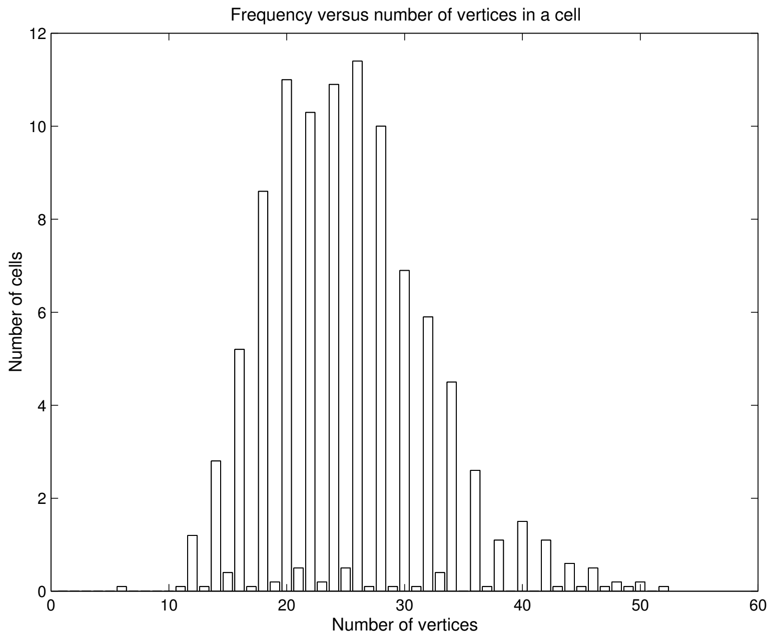
A programme submitted in January and killed in February due to its idleness had not gone to sleep as Nicholas suspected and Iain believed. The problem was with NQS. It has not been set up properly and it did not stop the process when after 24 hours. Instead, it let it run until well over 13 hours CPU time without doing anything about it. That process could, however, possibly have gradually slowed down for lack of resource, but not because diligence was missing. The people who looked after NQS obviously did not even realise this fact. Lack of time, as has been said, could not be a sound excuse since this was their major job.

Some interesting facts I found out about NQS from this experience are the following. The command `qdel` can kill only jobs waiting in a queue. A running job needs to be killed from the platform it is running. NQS needs to be set up properly in order for it to kill a process which reaches its time limit. Matlab batch files might not expire when run on NQS. This queueing system is very convenient except that it needs absolute pathing in everything you do. Is there any way to configure NQS or write a shell script to help making it more user-friendly?

§ More on number of vertices 3-d

The following was done on a 3–d systems with a joggled-input option in *qhull*. The number of vertices, however, takes into account the boundary cells as well. The minimum $n_{v,c}$ was 6, maximum $n_{v,c}$ 52, $\bar{n}_{v,c}$ 25.269, 2nd–Moment 49.363, 3rd–Moment 228.87, 4th–Moment 8,704.9, $\sigma^2_{n_{v,c}}$ 49.412, $\sigma_{n_{v,c}}$ 7.0294, $\mu_{g,n_{v,c}}$ 24.317, $\mu_{h,n_{v,c}}$ 23.365, $\widetilde{n}_{v,c}$ 24, $\delta_\mu(n_{v,c})$ 5.5428, $\kappa(n_{v,c})$ 3.5725 .

Following is a distribution graph.

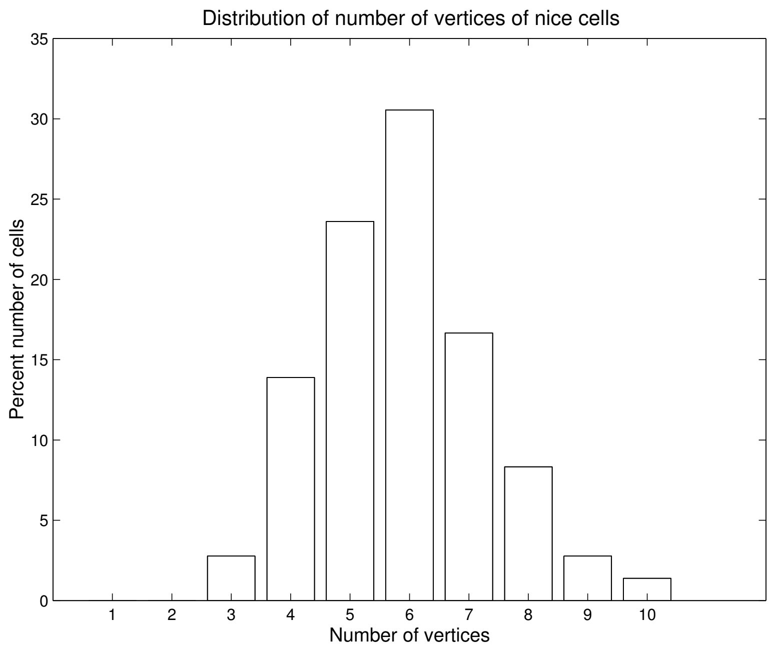


Faces in different dimensions

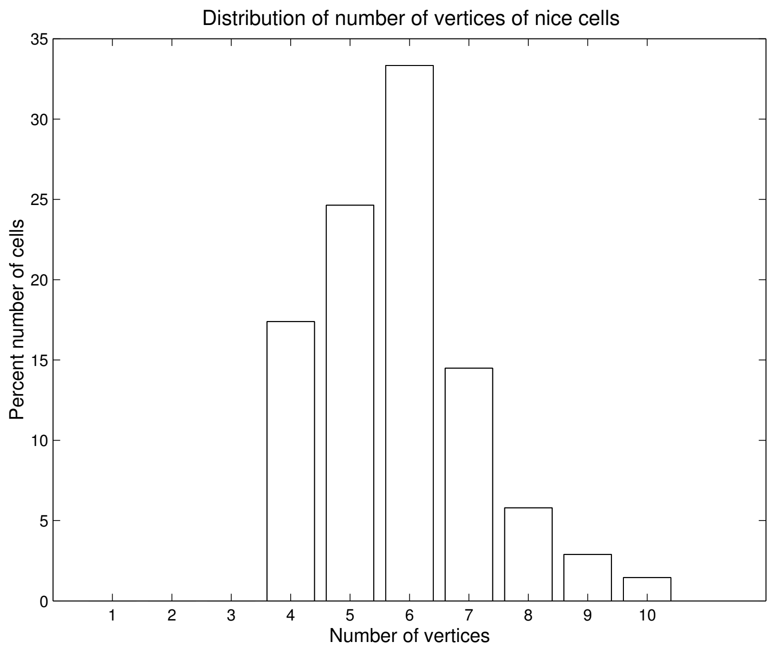
Considering only those cells bound withing the unit box, vertices and all, the following, namely Tables 6, 7 and 8, are the results from five simulations in two, three and four dimensions respectively, using the programme listed in Programmes section.

	Random seed				
	829247	134315	67453	432243	231215
N_v	187	187	186	189	183
n_v	170	167	168	163	170
n_c	72	69	70	65	72
n_c^v	169	165	166	159	170
μ_c^v	5.875	5.8116	5.8429	5.9077	5.8194
$(\sigma_v^2)_c$	2.0264	1.8022	1.6706	1.7726	1.5022
$m^2(n_c^v)$	1.9983	1.7761	1.6467	1.7453	1.4813
$m^3(n_c^v)$	1.1263	1.6917	0.96591	0.57642	0.96103
$n_{1\mathrm{f}}$	241	235	237	227	241
$n_c^{1\mathrm{f}}$	240	233	235	223	241
$\mathrm{t}_{\mathrm{CPU}}$(second)	20.36	20.39	20.09	20.6	19.76

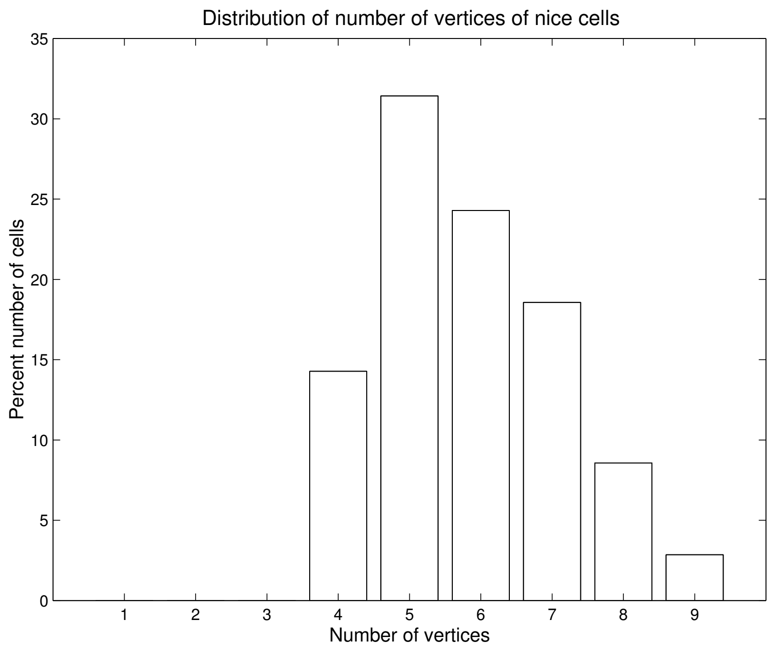
Table 6 *Faces of Voronoi in two dimensions.* $N_c = 100$.


Distribution of number of vertices of nice cells
Percent number of cells
Number of vertices
0
5
10
15
20
25
30
35
1
2
3
4
5
6
7
8
9
10

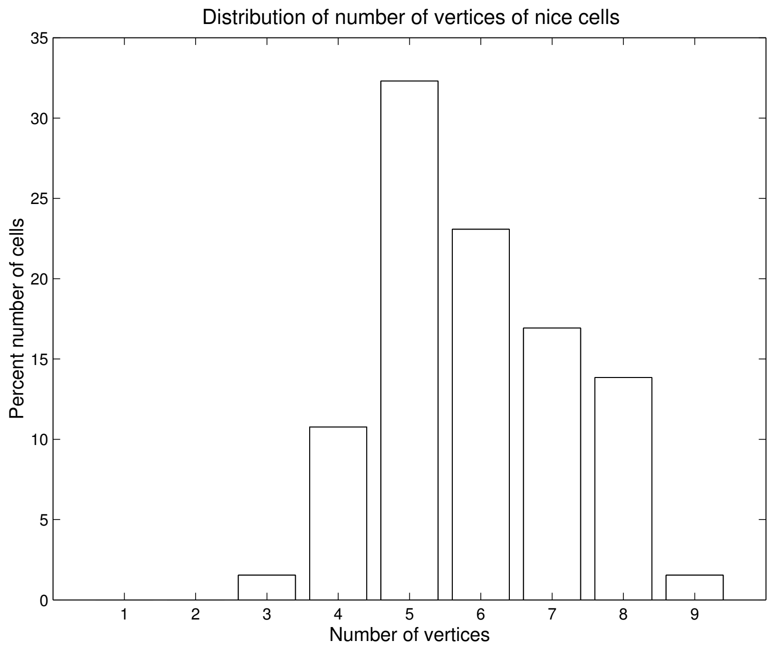
(a)


Distribution of number of vertices of nice cells
Percent number of cells
Number of vertices
0
5
10
15
20
25
30
35
1
2
3
4
5
6
7
8
9
10

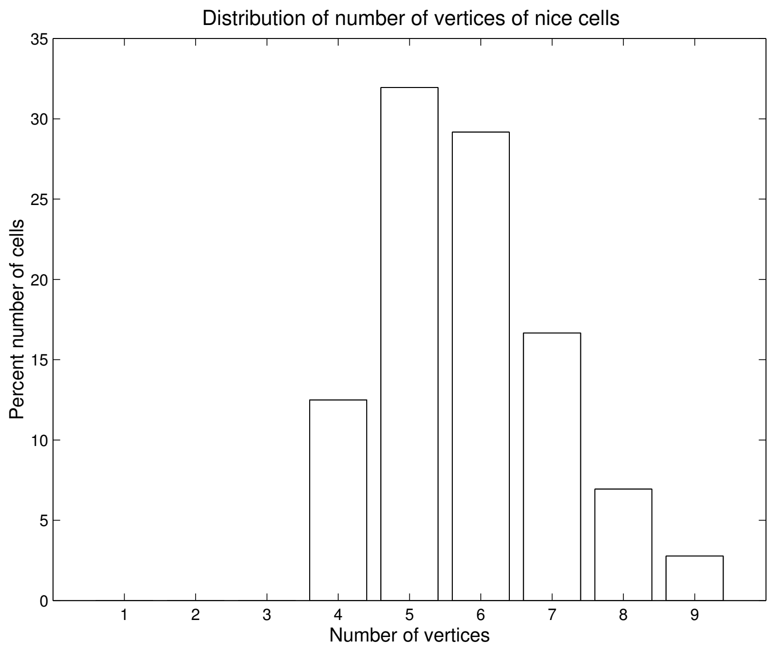
(b)


Distribution of number of vertices of nice cells
Percent number of cells
Number of vertices
0
5
10
15
20
25
30
35
1
2
3
4
5
6
7
8
9

(c)


Distribution of number of vertices of nice cells
Percent number of cells
Number of vertices
0
5
10
15
20
25
30
35
1
2
3
4
5
6
7
8
9

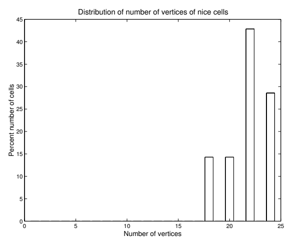
(d)



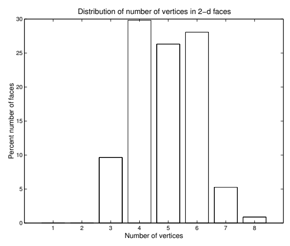
(e)

	Random seed				
	42398198	83250	34959	743690	1321
N_v	204	221	224	225	214
n_v	148	147	143	155	146
n_c	7	5	7	8	7
n_c^v	99	79	85	109	97
μ_c^v	21.714	23.6	20.286	21	21.714
$(\sigma_v^2)_c$	4.5714	14.8	31.238	34.286	21.905
$m^2(n_c^v)$	3.9184	11.84	26.776	30	18.776
$m^3(n_c^v)$	-4.6181	-16.128	-42.402	171	69.831
$n_{2\mathrm{f}}$	114	110	111	123	113
$\mu_{2\mathrm{f}}^v$	4.9211	4.7727	4.7928	4.8293	4.7788
$(\sigma_v^2)_{2\mathrm{f}}$	1.2592	1.6268	1.5476	2.0772	1.656
$m^2(n_{2\mathrm{f}}^v)$	1.2482	1.612	1.5336	2.0603	1.6413
$m^3(n_{2\mathrm{f}}^v)$	0.16453	0.99264	1.1334	1.7676	1.1444
$n_c^{2\mathrm{f}}$	76	60	69	88	75
$(\mu_{2\mathrm{f}}^v)_c$	5.0526	5.05	4.9855	4.9545	5.0533
$(\sigma_v^2)_c^{2\mathrm{f}}$	1.1439	1.811	1.3674	2.0209	1.7268
$m^2((n_{2\mathrm{f}}^v)_c)$	1.1288	1.7808	1.3476	1.9979	1.7038
$m^3((n_{2\mathrm{f}}^v)_c)$	0.19004	0.28275	1.5224	1.4544	0.98057
$n_c^{1\mathrm{f}}$	167	133	146	187	163
$\mathrm{t}_{\mathrm{CPU}}$(second)	24.28	25.67	28.49	34.23	26.46

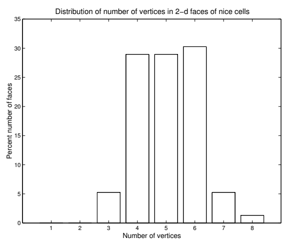
Table 7 *Various faces of Voronoi in three dimensions.* $N_c = 50$.



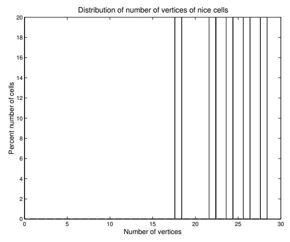
(a1)



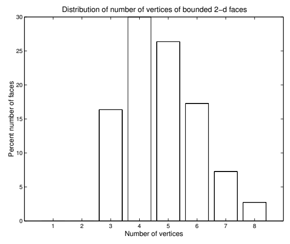
(b1)



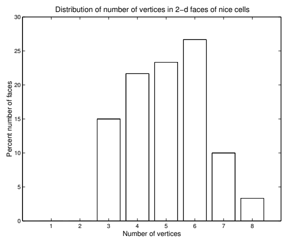
(c1)



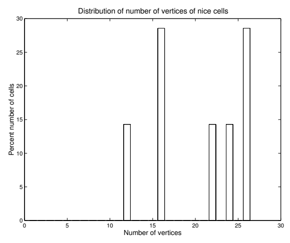
(a2)



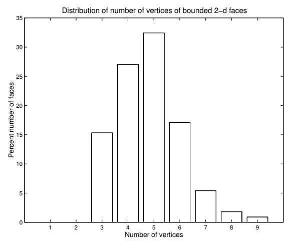
(b2)



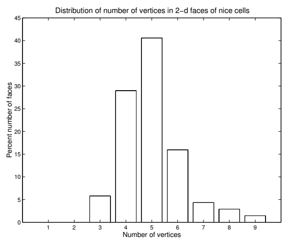
(c2)



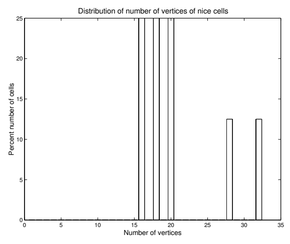
(a3)



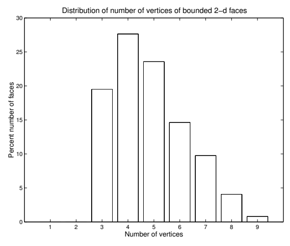
(b3)



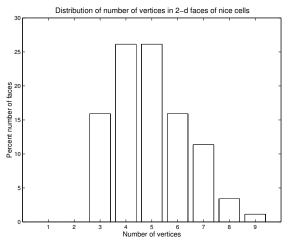
(c3)



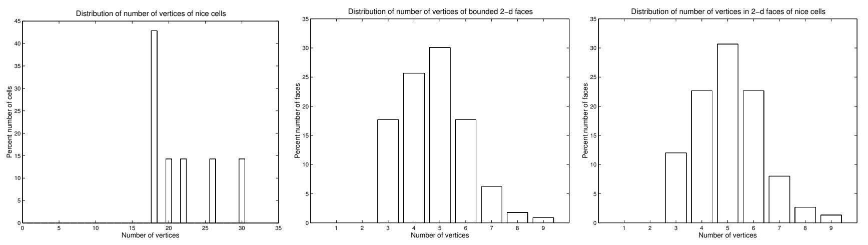
(a4)



(b4)



(c4)



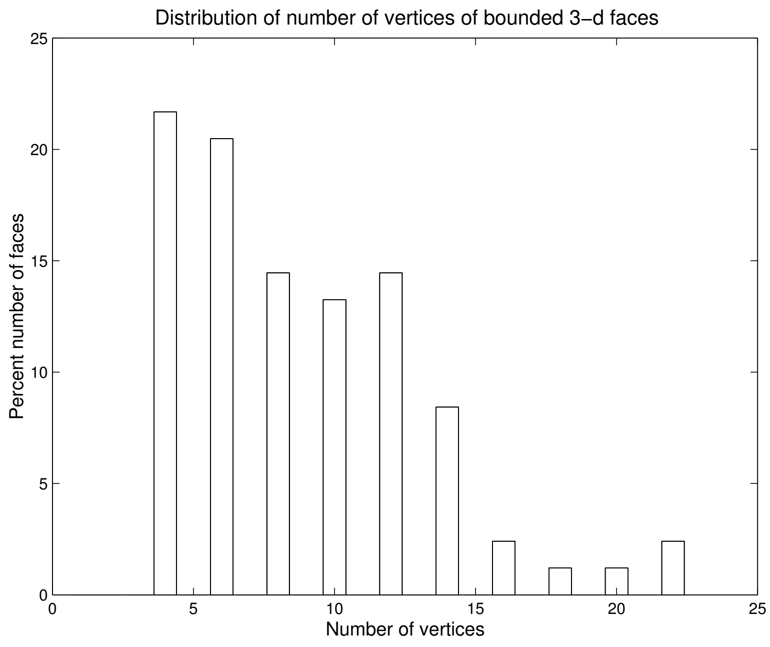
(a5) (b5) (c5)

Figure 36 *Distribution of* (ai) v_c, (bi) $v_{2\mathrm{f}}$, *and* (ci) $v_c^{2\mathrm{f}}$ *of the* i^{th} *simulation on 3-d Voronoi.*

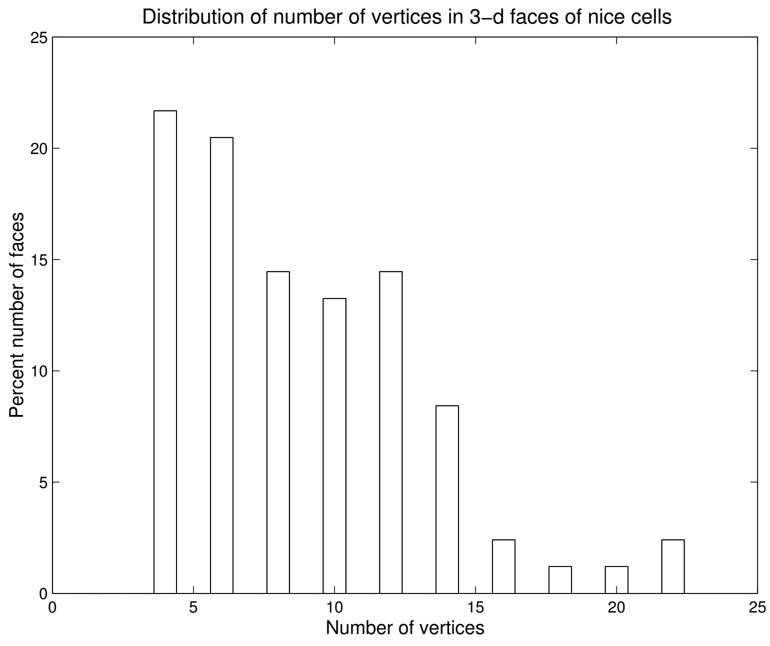
	Random seed				
	24098	802723	1453	732849	20480
N_v	1684	1719	1674	1586	1600
n_v	938	921	860	889	904
n_c	5	1	1	1	2
n_c^v	442	172	149	117	200
μ_c^v	114	172	149	117	110
$(\sigma_v^2)_c$	174.5	0	0	0	72
$m^2(n_c^v)$	139.6	0	0	0	36
$m^3(n_c^v)$	-547.2	0	0	0	0
$n_{3\mathrm{f}}$	83	102	84	83	95
$\mu_{3\mathrm{f}}^v$	8.8675	8.3137	8.5952	8.9398	8.7789
$(\sigma_v^2)_{3\mathrm{f}}$	18.848	18.198	19.449	17.496	19.77
$m^2(n_{3\mathrm{f}}^v)$	18.621	18.019	19.217	17.286	19.562
$m^3(n_{3\mathrm{f}}^v)$	75.225	102.38	95.854	62.919	118.61
$n_c^{3\mathrm{f}}$	32	8	10	7	19
$(\mu_{3\mathrm{f}}^v)_c$	8.8675	9.25	10.8	11.143	9.8947
$(\sigma_v^2)_c^{3\mathrm{f}}$	18.848	13.643	21.511	14.476	28.655
$m^2((n_{3\mathrm{f}}^v)_c)$	18.621	11.938	19.36	12.408	27.147
$m^3((n_{3\mathrm{f}}^v)_c)$	75.225	18.281	66.624	8.5364	195.1
$n_c^{2\mathrm{f}}$	11	0	5	2	7
$n_c^{1\mathrm{f}}$	2	0	0	0	1
$\mathrm{t_{CPU}}$(second)	315.07	358.94	324.56	281.85	283.82

Table 8 *Faces of Voronoi in four dimensions.* $\mathrm{N}_c = 100$.

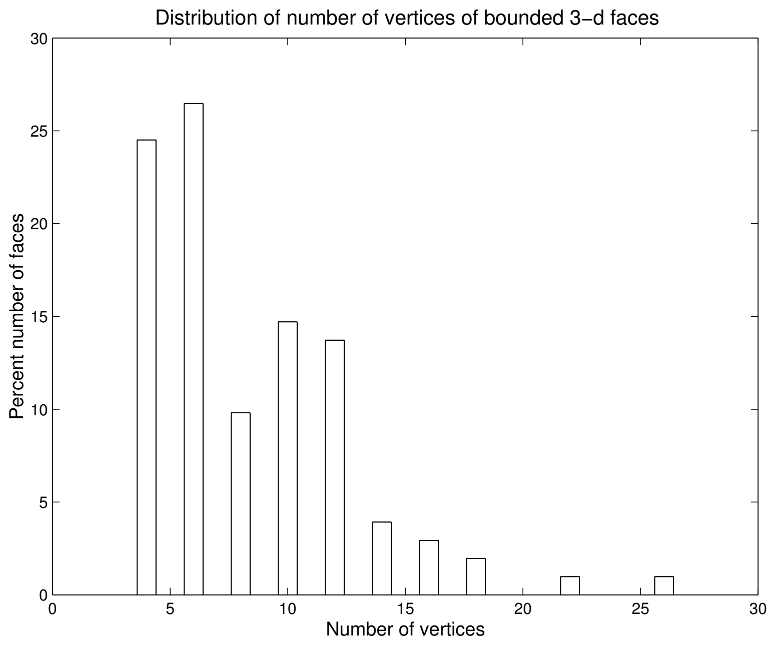
The number of vertices in each 2-d face is a constant equals to three while that of a 1-d face is two. In the first run the number of vertices in each of the cells is 95, 108, 117, 120 and 130; in the last run, this number is 104 and 116.


Distribution of number of vertices of bounded 3−d faces
Percent number of faces
Number of vertices
0
5
10
15
20
25

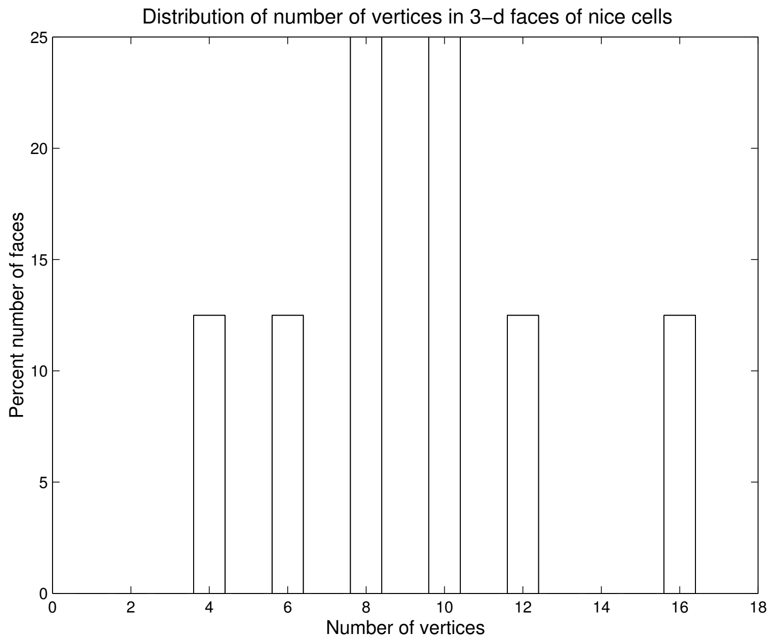
(a1)


Distribution of number of vertices in 3−d faces of nice cells
Percent number of faces
Number of vertices
0
5
10
15
20
25

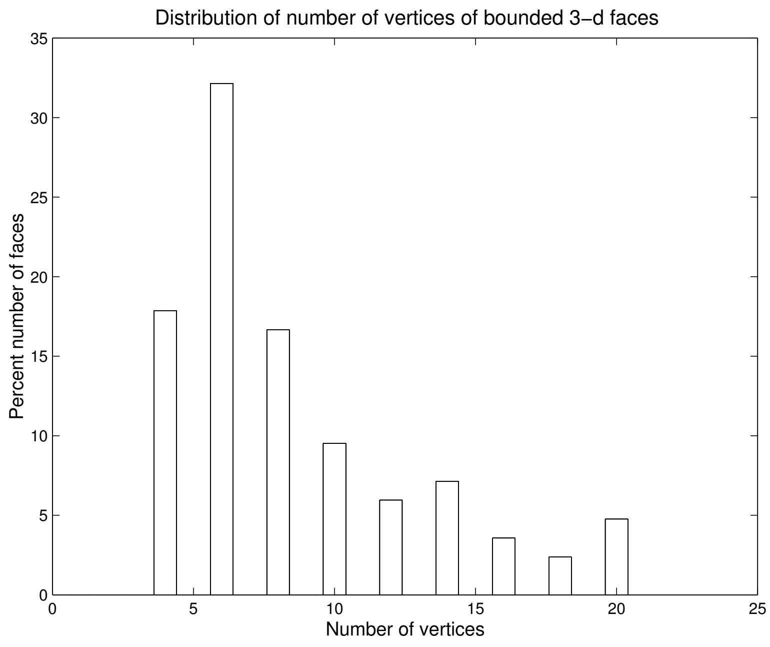
(b1)


Distribution of number of vertices of bounded 3−d faces
Percent number of faces
Number of vertices
0
5
10
15
20
25
30

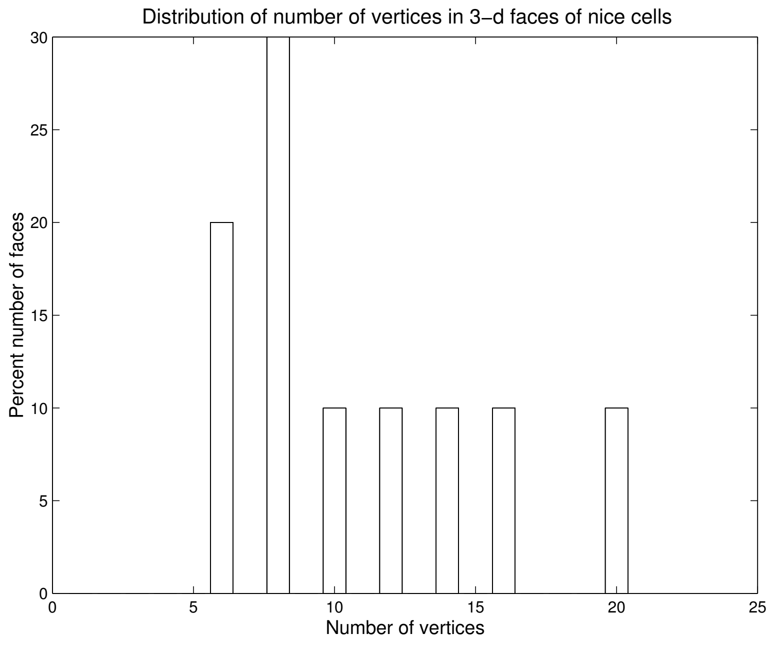
(a2)


Distribution of number of vertices in 3−d faces of nice cells
Percent number of faces
Number of vertices
0
2
4
6
8
10
12
14
16
18
5
15
20
25

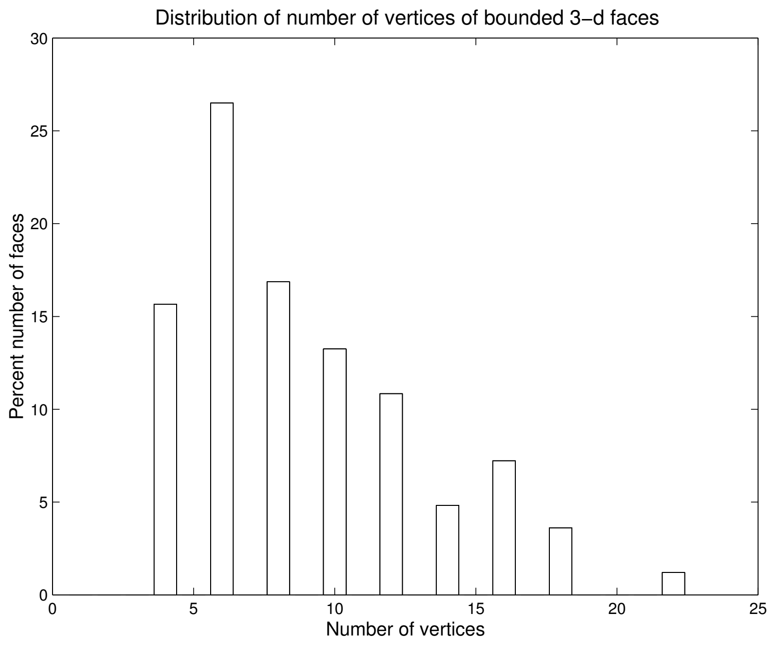
(b2)


Distribution of number of vertices of bounded 3−d faces
Percent number of faces
Number of vertices
0
5
10
15
20
25
30
35
0
5
10
15
20
25

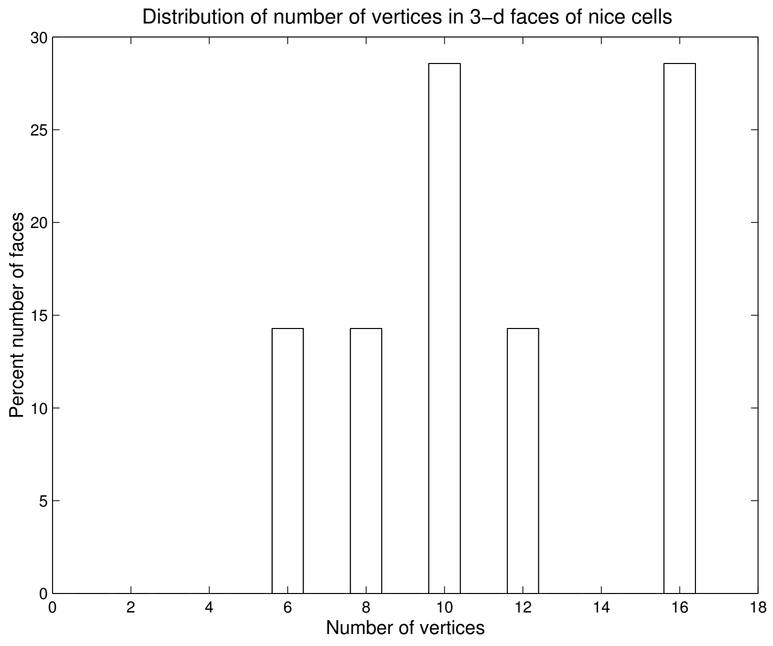
(a3)


Distribution of number of vertices in 3−d faces of nice cells
Percent number of faces
Number of vertices
0
5
10
15
20
25
30
0
5
10
15
20
25

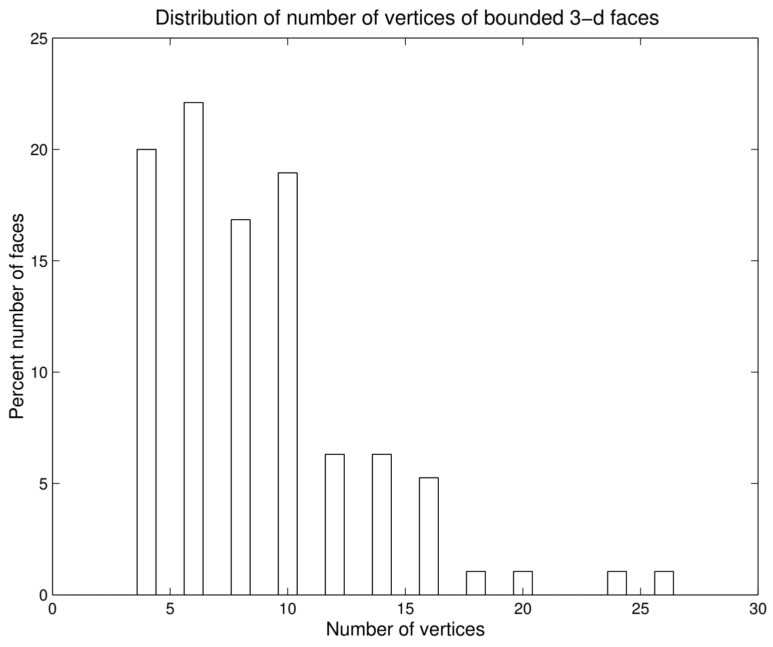
(b3)


Distribution of number of vertices of bounded 3−d faces
Percent number of faces
Number of vertices
0
5
10
15
20
25
30
0
5
10
15
20
25

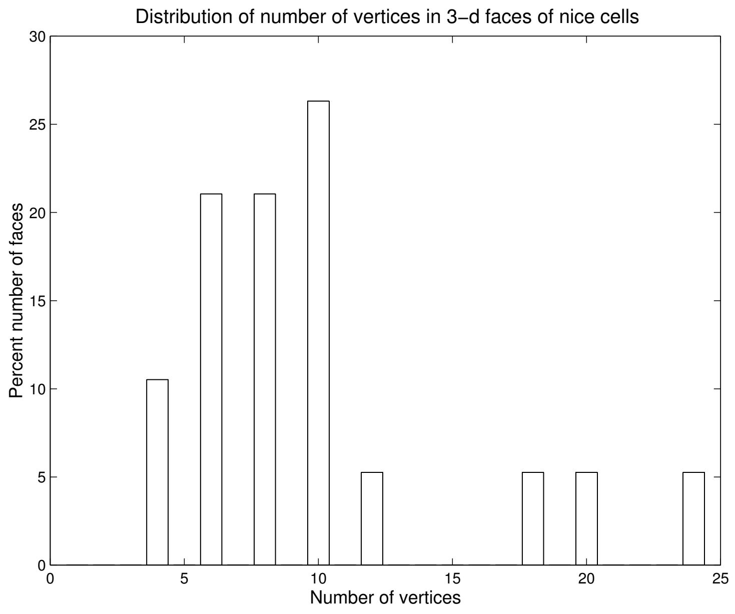
(a4)


Distribution of number of vertices in 3−d faces of nice cells
Percent number of faces
Number of vertices
0
5
10
15
20
25
30
0
2
4
6
8
10
12
14
16
18

(b4)



(a5)



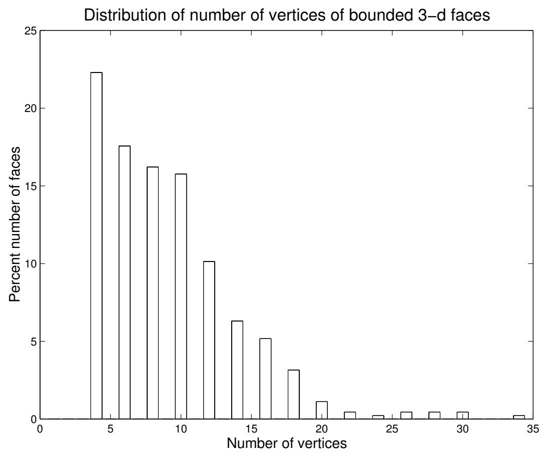
(b5)

Figure 37 *Distribution of* (ai) $v_{3\mathrm{f}}$ *and* (bi) $v_c^{3\mathrm{f}}$ *of the* i^{th} *simulation on 4-d Voronoi.*

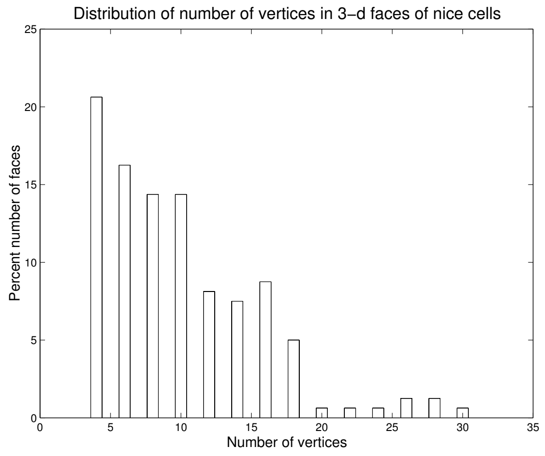
Table 9 contains the results obtained from a 4-d Voronoi network of 300 original nuclei. The numbers of vertices of the sixteen cells are 97, 99, 112, 141, 145, 160, 170, 171, 176, 176, 184, 186, 188, 192, 216 and 235.

Random seed	91876				
N_v	6776	$m^3(n_c^v)$	-1.7507×10^4	$(\mu_{3f}^v)_c$	9.9875
n_v	3848	n_{3f}	444	$(\sigma_v^2)_c^{3f}$	31.1697
n_c	16	μ_{3f}^v	9.1486	$m^2((n_{3f}^v)_c)$	30.9748
n_c^v	1687	$(\sigma_v^2)_{3f}$	24.0411	$m^3((n_{3f}^v)_c)$	203.3116
μ_c^v	165.5000	$m^2(n_{3f}^v)$	23.9869	n_c^{2f}	60
$(\sigma_v^2)_c$	1.5100×10^3	$m^3(n_{3f}^v)$	170.4755	n_c^{1f}	3
$m^2(n_c^v)$	1.4156×10^3	n_c^{3f}	160	t_{CPU}(second)	1.6013×10^4

Table 9 *Face statistics of 300 nuclei Voronoi in four dimensions.*



(a)



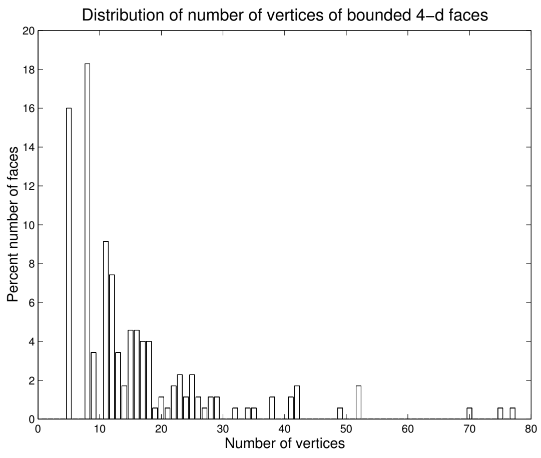
(b)

Figure 38 *Distribution of* (a) $v_{3\mathrm{f}}$ *and* (bi) $v_c^{3\mathrm{f}}$ *of 4-d Voronoi, 300 nuclei.*

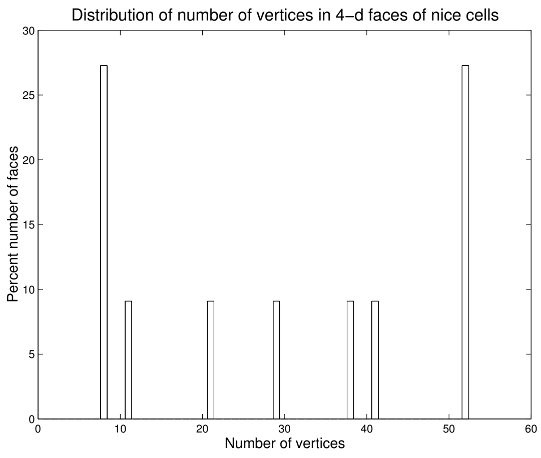
Table 10 is obtained from Voronoi in five dimensions.

Random seed	39378			N_v	16212
n_v	6449	$n_{4\mathrm{f}}$	175	$(\sigma_v^2)_c^{4\mathrm{f}}$	352.2909
n_c	1	$\mu_{4\mathrm{f}}^v$	15.9200	$m^2((n_{4\mathrm{f}}^v)_c)$	320.2645
n_c^v	864	$(\sigma_v^2)_{4\mathrm{f}}$	163.7752	$m^3((n_{4\mathrm{f}}^v)_c)$	351.8362
μ_c^v	864	$m^2(n_{4\mathrm{f}}^v)$	162.8393	$n_c^{3\mathrm{f}}$	1
$(\sigma_v^2)_c$	0	$m^3(n_{4\mathrm{f}}^v)$	4.8764×10^3	$(\mu_{3\mathrm{f}}^v)_c$	4
$m^2(n_c^v)$	0	$n_c^{4\mathrm{f}}$	11	$n_c^{2\mathrm{f}}$	1
$m^3(n_c^v)$	0	$(\mu_{4\mathrm{f}}^v)_c$	29.0909	$\mathrm{t}_{\mathrm{CPU}}$(second)	1.8908×10^4

Table 10 *Face statistics of 200 nuclei Voronoi in five dimensions.*



(a)

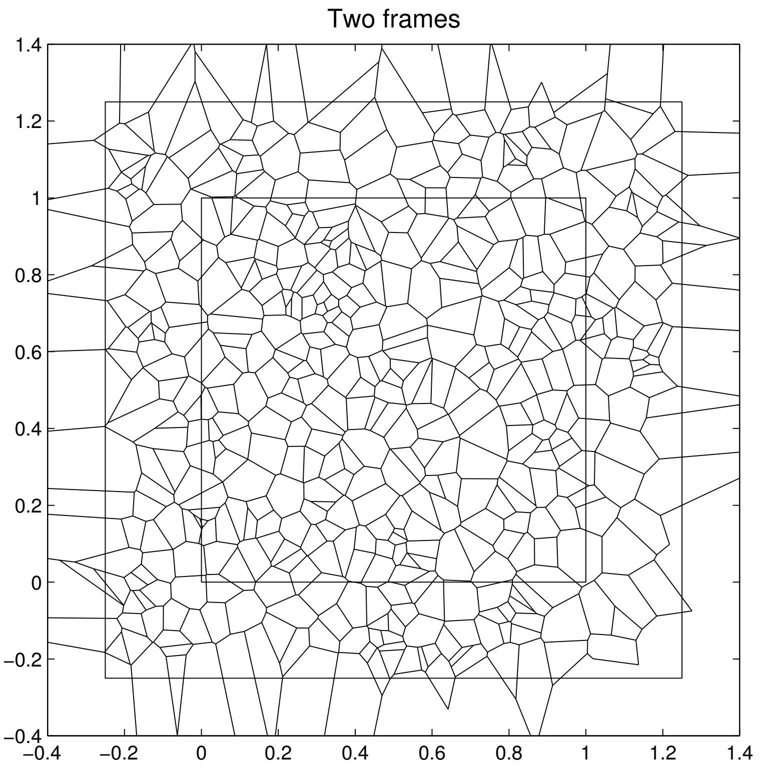


(b)

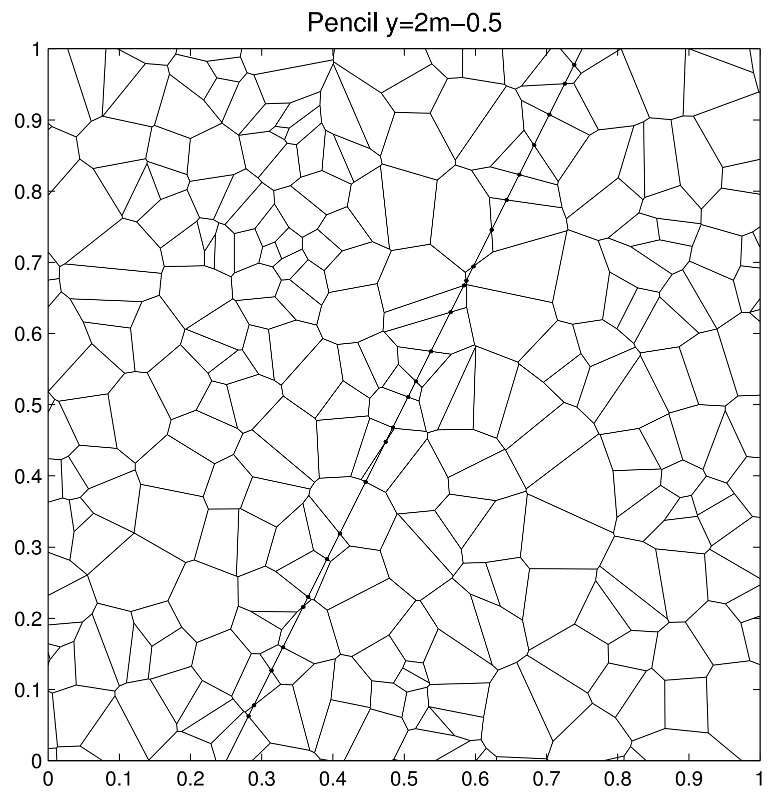
Figure 39 *Distribution of* (a) $v_{4\text{f}}$ *and* (bi) $v_c^{4\text{f}}$ *of 5-d Voronoi, 200 nuclei.*

Beam intersection study

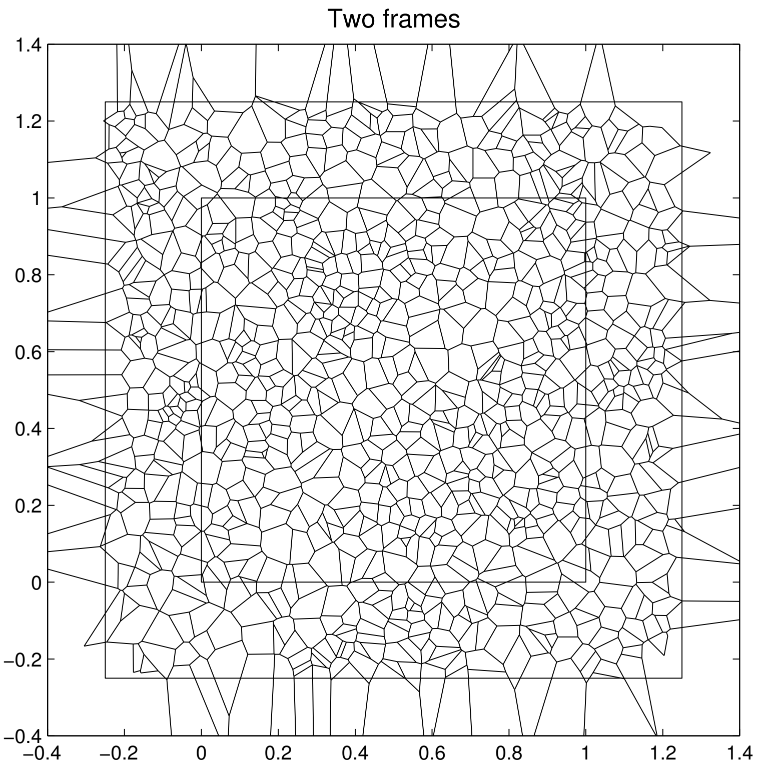
In this study of sectioning by a line the Voronoi in two dimensions, first generates on Matlab 500 points within a square box from -0.25 to 1.25 in both axes. The beam is simply a straight line $y = mx + c$ where m is the slope and c a constant.



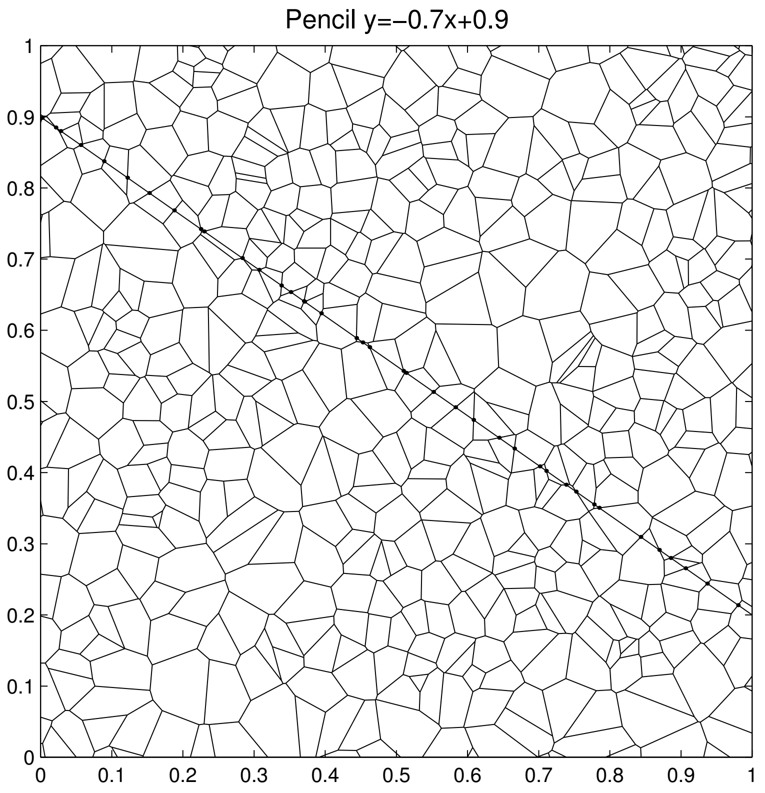
(a)



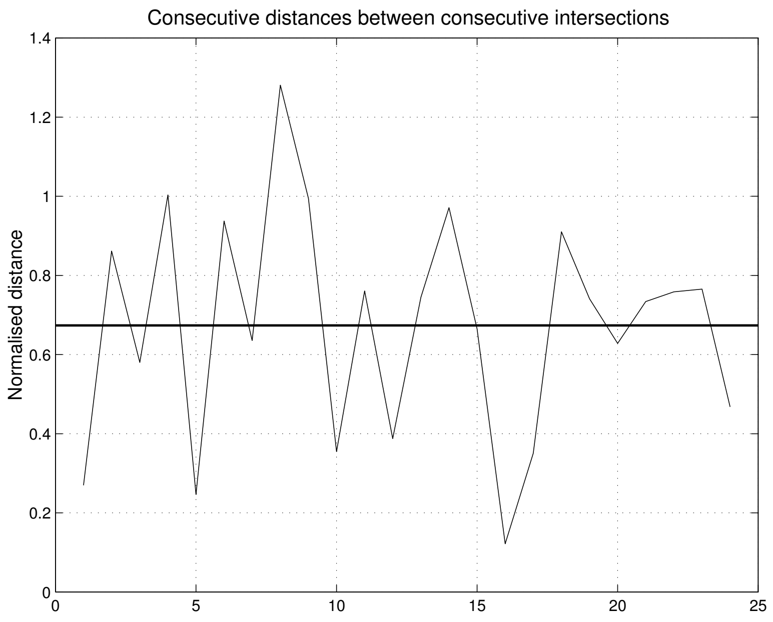
(b)



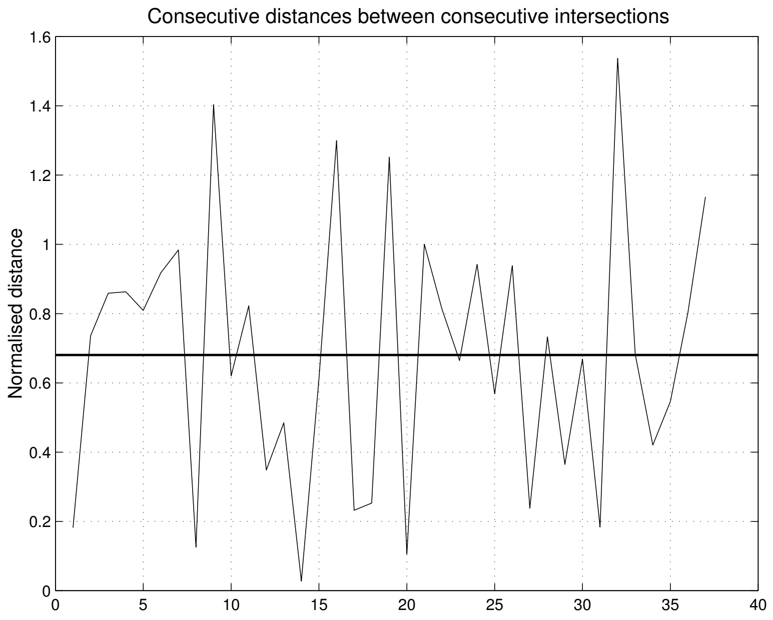
(c)



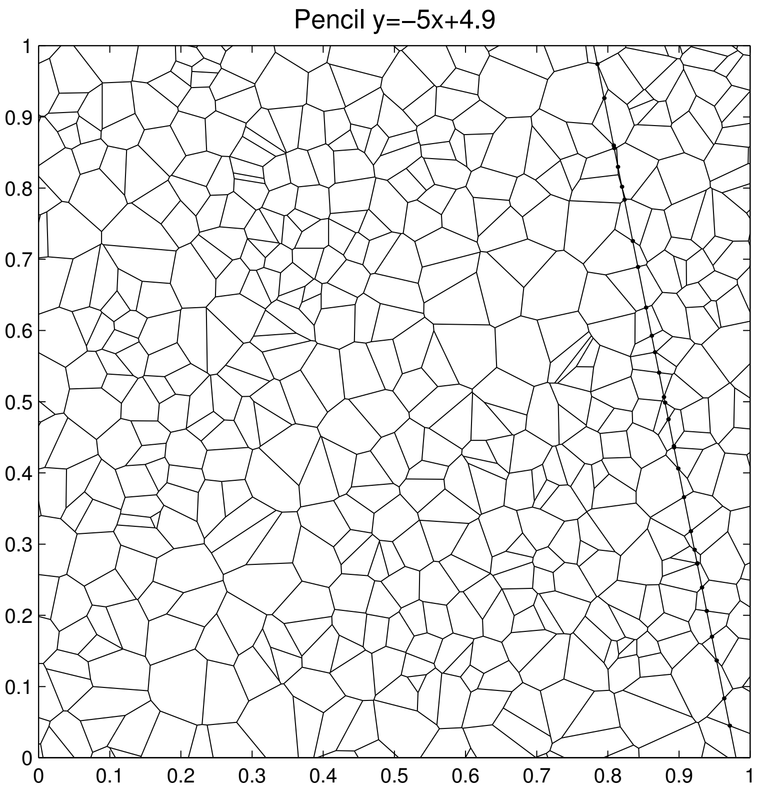
(d)


Consecutive distances between consecutive intersections
Normalised distance
0
0.2
0.4
0.6
0.8
1
1.2
1.4
0
5
10
15
20
25

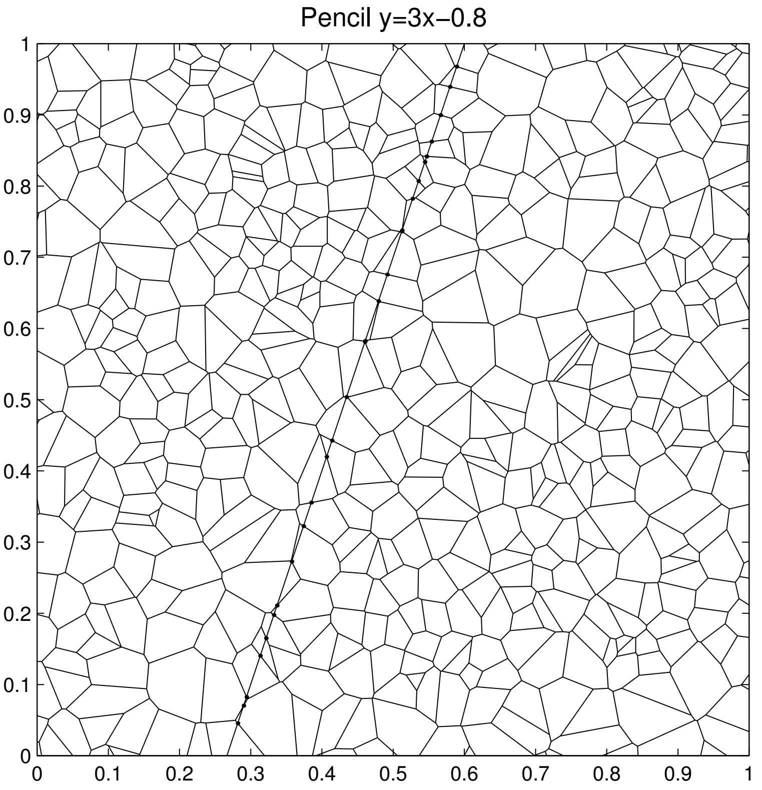
(e)


Consecutive distances between consecutive intersections
Normalised distance
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
0
5
10
15
20
25
30
35
40

(f)



(g)



(h)

Figure 11 *Intersection by a line.* (a) *is intersected by* (b) $y = 2x - 0.5$*;* (c) *is intersected by* (d) $y = -0.7x + 0.9$*;* (e) *and* (f) *are corresponding distances of respectively* (b) *and* (d)*;* (c) *is intersected by* (g) $y = -5x + 4.9$ *and* (h) $y = 3x - 0.8$*.*

A natural basis for the normalisation is the expected distance. Another possible basis is $\frac{1}{\sqrt{239}} = 0.064685$. I call *mean normalisation* the normalisation using the first basis and *homogeneity normalisation* the second. Graphs of the closest pair distances look the same for both types of normalisation.

Simulation	1	2	3	4
N_c	500	1,000		
N_e	1,463	2,955		
n_c	239	463		
n_e	789	1,455		
Line equation	$y = 2x - 0.5$	$y = -0.7x + 0.9$	$y = -5x + 4.9$	$y = 3x - 0.8$
$\bar{d}$	4.6867×10^{-2}	3.163×10^{-2}	3.2448×10^{-2}	3.7843×10^{-2}
σ_d^2	3.9422×10^{-4}	3.1174×10^{-4}	2.9005×10^{-4}	4.9535×10^{-4}
Normalisation				
by mean	1 ± 0.4237 0.1713 -8.1332×10^{-3}	1 ± 0.5582 0.3032 3.5783×10^{-2}	1 ± 0.5249 0.2656 -3.7762×10^{-3}	1 ± 0.5881 0.3321 2.2989×10^{-2}
by homogeneity	0.7245 ± 0.3070 8.9936×10^{-2} -3.0936×10^{-3}	0.68059 ± 0.3799 0.1404 1.128×10^{-2}	0.6982 ± 0.3665 0.1295 -1.2853×10^{-3}	0.81428 ± 0.4789 0.2202 1.2412×10^{-2}

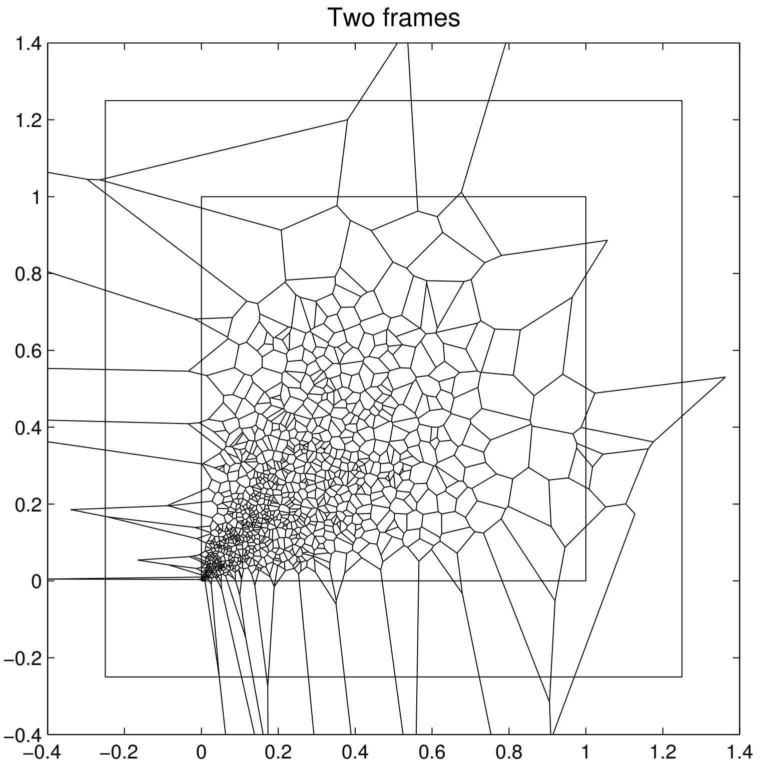
The programme used is listed at the end. The space position vector of the intersection between the two vectors AB and CD is $P = A + r(B -$

A), where $AB = B - A$ and $CD = D - C$ are vectors or directed lines and A, B, C, D are space vectors, $r = \begin{vmatrix} CD' \\ CA' \end{vmatrix} \Big/ \begin{vmatrix} AB' \\ CD' \end{vmatrix}$ and $s = \begin{vmatrix} AB' \\ CA' \end{vmatrix} \Big/ \begin{vmatrix} AB' \\ CD' \end{vmatrix}$ Here $AB = A + r(B - A)$, and $CD = C + s(D - C), 0 \leq r, s \leq 1$ are directed lines. P exists if $0 \leq r \leq 1$ and $0 \leq s \leq 1$.

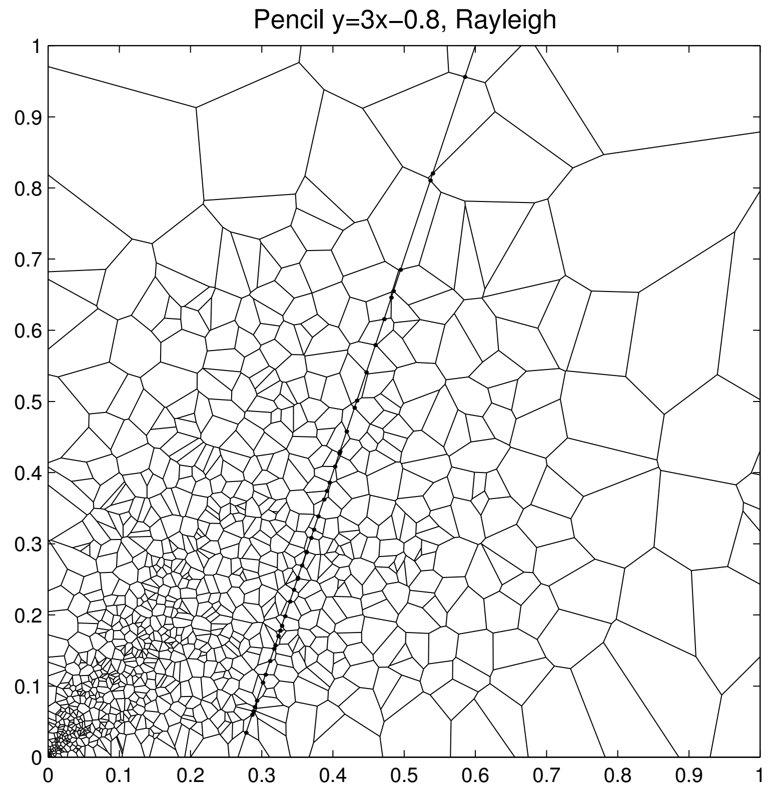
If the denominator $\begin{vmatrix} AB' \\ CD' \end{vmatrix}$ is zero, then the two lines are parallel. Also, if the nominator of r is zero, that is $\begin{vmatrix} CD' \\ CA' \end{vmatrix} = 0$, then both lines are collinear.

Consider the line section of Rayleigh distributed Voronoi where both the coordinates x and y are random numbers with Rayleigh distribution. The probability density function of the Rayleigh distribution is $y = f(x/b) = \frac{x}{b^2} e^{\frac{-x^2}{2b^2}}$. The mean of this distribution is $b\sqrt{\frac{\pi}{2}}$ and the variance is $\frac{4-\pi}{2} b^2$. With $b = 1, 2, \ldots, 1000$ choose the random numbers from the Rayleigh distribution, then scale and use them as the coordinates.

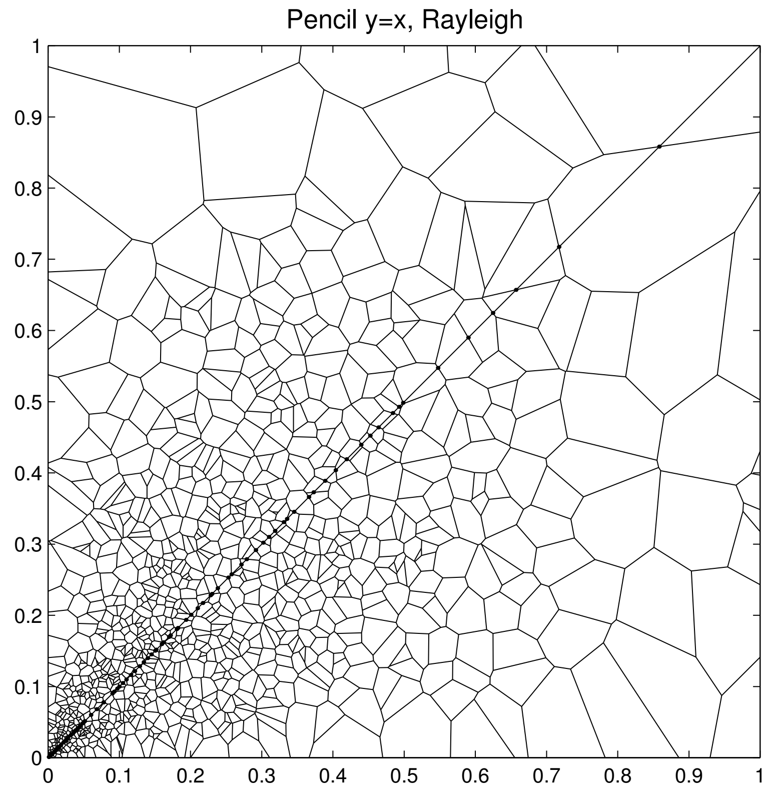
In Figure 41 the structure in (a) is intersected by (b) $y = 3x - 0.8$, (c) $y = x$, (d) $y = 2x$, (e) $y = 0.2x$, (f) $y = -x + 1$, (g) $y = -x + 0.3$.



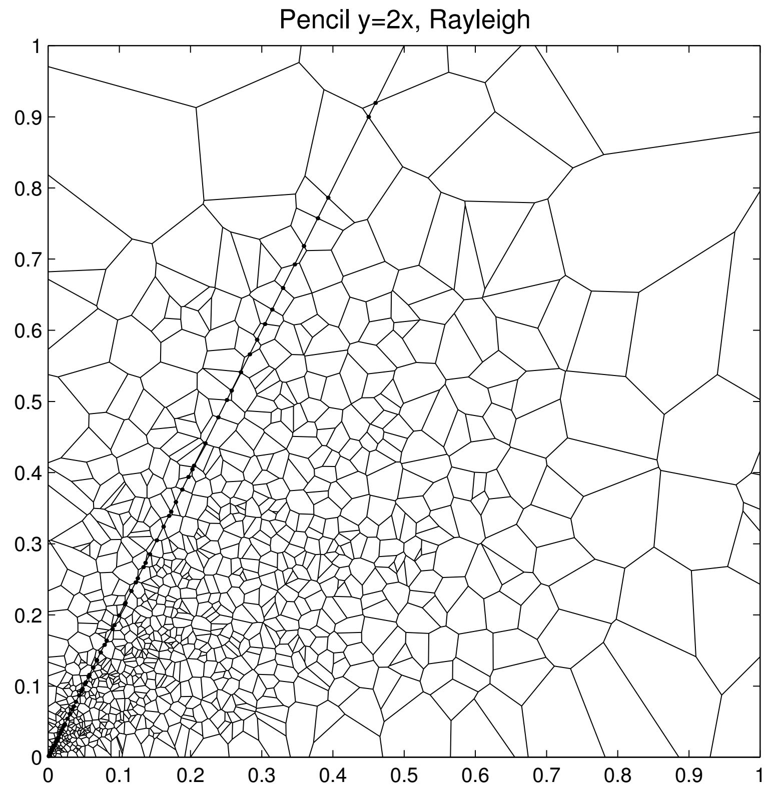
(a)



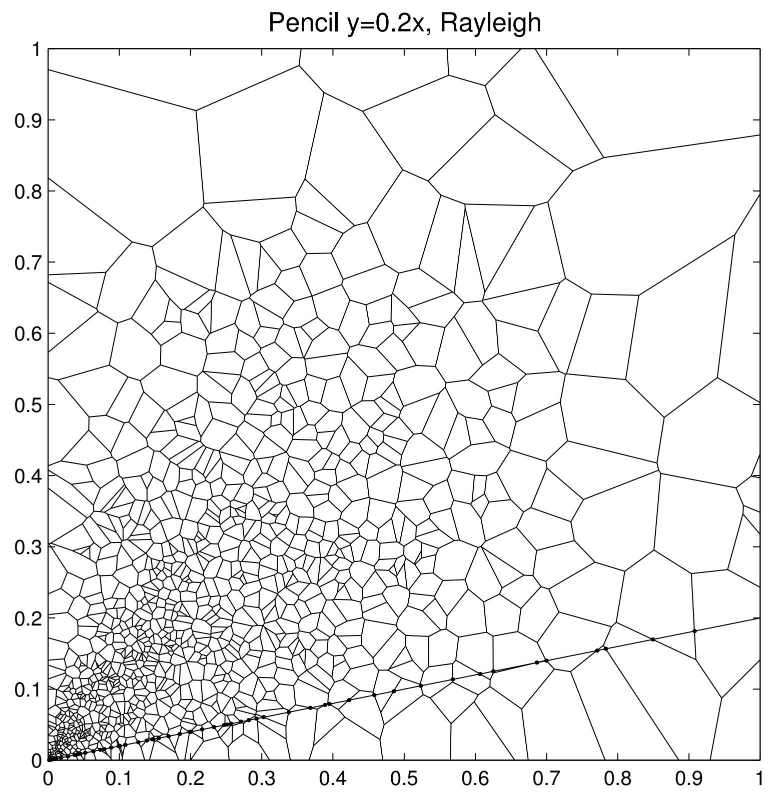
(b)



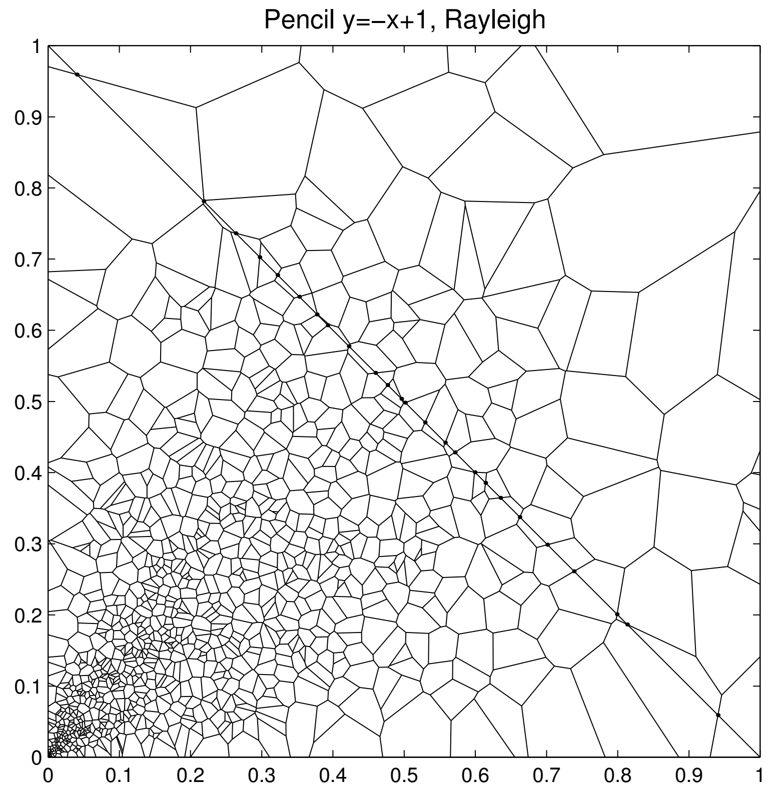
(c)



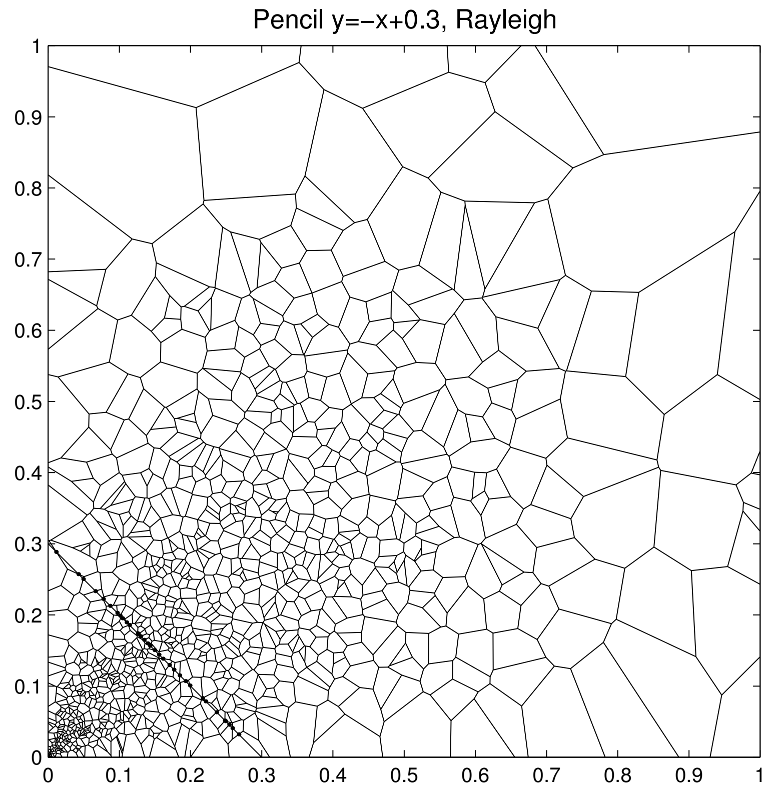
(d)


Pencil y=0.2x, Rayleigh
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

(e)



(f)



(g)

Simulation	1	2	3	4	5	
N_c	1,000					
N_e	2,975					
n_c	988					
n_e	2,912					
Line eq.	$y = 3x - 0.8$	$y = x$	$y = 2x$	$y = 0.2x$	$y = -x + 1$	$y = -x + 0.3$
$\bar{d}$	2.3621×10^{-2}	1.4253×10^{-2}	1.3853×10^{-2}	1.7434×10^{-2}	4.4401×10^{-2}	1.027×10^{-2}
σ_d^2	8.1582×10^{-4}	6.2922×10^{-4}	2.9647×10^{-4}	3.0991×10^{-4}	1.1584×10^{-3}	3.5365×10^{-5}
Normalised						
by mean	1 ± 1.2092	1 ± 1.7599	1 ± 1.2429	1 ± 1.0098	1 ± 0.7665	1 ± 0.5791
	1.4256	3.0609	1.5239	1.0004	0.56204	0.3245
	5.5516	28.351	7.6499	1.6643	1.2114	0.1755
by homog.	0.7425 ± 0.8978	0.4480 ± 0.7885	0.4354 ± 0.5412	0.548 ± 0.5533	1.3956 ± 1.0698	0.3228 ± 0.1869
	0.7859	0.6144	0.2890	0.3004	1.0947	3.3813×10^{-2}
	2.2721	2.5493	0.6316	0.2739	3.293	5.9021×10^{-3}

Table 12 *Line intersection statistics of Rayleigh distributed Voronoi*

The following codes generate and scale the points of Rayleigh distribution.

```
X=raylrnd([1:NumCell])';
Y=raylrnd([1:NumCell])';
Max=0.8*max([X;Y]);
X=X/Max;
Y=Y/Max;
```

The values of Cx and Dx will need to be adjusted manually from the pencil beam equation. It is the value of x at both points of intersection between the beam and the $[0, 1]$ square box. A random Voronoi network can either be both homogeneous and isotropic or, nonhomogeneous and nonisotropic depending on whether the probability distribution function is a constant.

Percolation

The 1970's and the 1980's saw a proliferation of variations on the theme of percolation. Every year there seemed to be a new percolation problem or two. For a *Bethe lattice* Chalupa *et al* (1979) reported a *bootstrap percolation* where those randomly occupied sites with less than m occupied neighbours are recursively emptied one by one until a stable configuration is reached. The problem they are interested in is that where the impurity concentration, dilution and crystal-field interaction compete in magnetic materials compete against the exchange interaction, resulting in the magnetic moments and consequently the magnetic order being destroyed.

Wilkinson and Willemsen (1983) introduced *invasion percolation*. Working with Schlumberger they were interested in the real problem in the oil industry where water displaces oil via capillary action. Their approach was that of a constant flow rate, not one of a constant pressure as usual. Here water displaces oil in the smallest available pores. But when water completely surrounds any region of oil, no further advance into that region will be possible, oil being incompressible. Such regions are called *trappings* and cause a problem generally known by the name of *residual oil*, an economic bane for oil industry.

In the above example the hydrophobic versus hydrophilic property plays an important role in the replacement of oil in pores with water. And water is prevented from penetrating trapped oil regions by much the same principle as that which prevents the water in the contents of a sandwich from crossing the spread layer of butter to wet the hydrophilic bread. For this latter case the pores in question are those within the bread texture, and the soaking of the bread is best prevented provided that the trapped regions of butter or margarine percolates in two dimensions to form a single layer or film which entirely covers the sectioning surface of the bread. Moreover, there is a similarity also in the internal structure of both the rock and the bread. The density of pores in bread is not homogeneous as a result of the tension on the surface of the dough caused by internal air pressure originated from the yeasts inside, as well as because of the heat applied to it when inside the oven, which dehydrolises the surface and makes it dry and hardened. The same inhomogeneity can be found inside the rocks which form the oil reservoirs where the regions of oil are surrounded by rocks which are less porous and have less permeability.

For Adler and Aharony (1988) a random walker, aka ant, treads on clusters. The ant enlarges a cluster by stepping on to an empty site next to it which meets certain conditions. They called this problem *diffusion percolation*. An example of a condition met is where the empty site has two or more occupied neighbours on a square lattice.

Most percolation in studies happens by randomly toggling the phases of sites or bonds in regular lattices. Kerstein (1983) considered randomly located spheres, take the complementary region of their union, and then perform percolation on the former. He showed that such percolation problem is equivalent to a percolation on a Voronoi lattice whose sites are the sphere centres.

In the same way as an infinite loop in computer science means that one can not come to the termination of a programme going along the time dimension, an infinite cluster in percolation means that one can not come to the end of a cluster shifting along either one of the dimensions. The former case is along the one dimension of time and is only possible because of the time flows in one direction. Therefore half the line spanned is considered as being infinite. In one nondirectional dimension, except for the trivial case where one can consider the entire space as being one single cluster, no infinite cluster is possible. In simulation, when a cluster spans the whole of the space considered along any dimension we say that it is infinite since as one moves a cross section along that dimension, it always contains a section of the cluster.

The bond percolation programme of two-dimensional Voronoi network by Tiyapan (1995, KNT4(iii), p. 78) takes $O(n^2)$ time provided we assume that the contribution made by the number of clusters together with the that by each cluster comparison amount to a linear time complexity term. In order to justify this assumption, consider first the number of clusters. The maximum number of clusters possible depends on the size of the system, in other words it must vary as n_1^k, where $0 < k_1 < 1$. This maximum value should be approximately 0.5 because of symmetry between occupancy and void clusters. Next consider the time involved in comparing two clusters. On Matlab this is a sparse vector comparison which is likely to involves some linked lists, and similarly should take time as a function of n_2^k, $0 < k_2 < 1$. Because on average the size of clusters is always small before Pc, k_2 will be less than 0.5. Therefore $n_1^k \cdot n_2^k < n$ and it is safe to assume $\mathrm{O}(n)$ time from both of them combined. *q.e.d.*. A C translation (Tiyapan 1995, *ibid.*, p. 80) of this programme, though not as bad as it may seem because constant coefficients are small, gives $O(n^5)$ time in comparison.

In the field of geology Miller *et al* (2000) studies the analogy between the dilatant slip in earthquakes and the hydrofraction occurred in melting and dehydration, the percolation of the latter in the permeability network internal to the fault being the cause of the former. According to them, the existence of toggle switches in the permeability rules out the assumption that the permeability throughout the whole system is homogeneous. Instead, the system reorganises itself into sytems of different scales of interaction according to the degree and nature of its inhomogeneity. At the critical state the scale of interaction is equal to the scale of the model.

The percolation probabilities given by Stauffer and Aharony (1994) are, the first number being for sites and the second for bonds, for the honey comb lattice 0.6962, 0.65271; square 0.592746, 0.50000; triangular 0.50000, 0.34729; diamond 0.43, 0.388; simple cubic 0.3116, 0.2488; body-centred cubic 0.246, 0.1803; face-centred cubic 0.198, 0.119; 4-d hypercubic 0.197, 0.1601; 5-d hypercubic 0.141, 0.1182; 6-d hypercubic 0.107, 0.0942; 7-d hypercubic 0.089, 0.0787.

De Gennes *et al* (1959) investigated disordered binary solid solution AB where active atoms A are randomly replace the nodes of the periodic matrix B. There exists a critical concentration A in B below which all clusters are finite, and above which both finite and non-finite, *i.e.* infinite, clusters exist. Such solid solution in networks can represent the spin waves in alloys with one ferromagnetic component or the impurity bands in semiconductors. They cited seminal work on percolation by Broadbent *et al* (1957), but no mention was made about the Ising model.

In a way, percolation is similar to diffusion. In diffusion the particles considered move about randomly, whereas in percolation they can only crop up randomly at predetermined locations on a network which is fixed. We could imagine, for instance, cars running along the roads within a traffic network as diffusing through them. Then the percolation could occur on a larger scale, that is the scale of a road. The cars move along, that means they diffuse; but the roads remain fixed, and so their phases could percolate. In other words, in diffusion the particles move while in percolation, whether there are moving particles or not, it is the phases that percolate. Since historically percolation began as the study of diffusion of particles in a network of tubes in which the phases are naturally defined as the tubes being blocked or unblocked, these definitions have become most frequently used in other areas of application, for example in filtering membranes and traffic networks.

But this is not necessarily the case. Instead of dealing with a fixed network, one may consider a model of percolation in a continuum, for example by randomly patching an area until all the patches connect with one another somehow and percolate. The patch could be of any shape, as well as polygonal and circular. We can consider the percolation in a certain area as having occurred when there appears a cluster of patches which traverses any two opposite sides. One application of this is in the study of occurrences of epidemics. Hoyle and Wickramasinghe (1979), having given a convincing argument in favour of viruses and various forms of diseases being carried to Earth from space by comets, talk about patchiness of pathogenic clouds. According to them, simultaneous attacks across vast region rules out person-to-person theory. Moreover, influenza epidemics are generally characterised by sudden onsets and equally sudden ends. These epidemics and plagues may be thought of as the percolation of these patches in a sufficiently large, predefined area. The bacteria and viruses coming from space adding to gene give the possibility of jump patterns in evolution (*cf* Smith and Szathmáry, 1995). The cells deliberately refuse to block viruses because they could prove to be useful in the long run, generally not by individuals but by the species. Historical examples are the disease described by Thucydides between 431 and 404 BC, five epidemics of 'English Sweats' between 1485 and 1552, and more than ten influenza pandemic from 1700 to 1900. Example of diseases which are caused by bacteria and viruses are bubonic plague, chicken pox (varicella), cholera, common cold, Legionnaires' disease, leprosy, measles, mumps, poliomyelitis, small pox, tuberculosis, and trachoma. Examples of major evolutionary transitions are those going from RNA to DNA, from prokaryotes to eukaryotes, and from asexual cloning to sexual propagation. Another example is the transition from primate to human both of whom differ from each other neurologically in the ability to use language and the power to conceptualise. One description of the Great Plague in London (Dickens, 1851). There was a rumour that a few people died in the winter of 1664. In May 1665 the disease burst out in St. Giles's which raged through July and September in every part of England; approximately 10,000 people died in London alone. But then the equinox winds virtually blew the disease away, and the Plague quickly disappeared. The existence of interstellar organic matters is supported by strong evidences with more and more complex substances constantly found (*cf* Hoyle and Wickramasinghe, 1978).

The growth mechanism of clusters in percolation is all so found in biology. Williams and Bjerknes (1972) simulate a tumour in the basal layer of an epithelium. The basal cells become less sensitive to Charlone which controls cellular division, and thus they divide κ times faster than the normal cells where $\kappa > 1$ is the carcinogenic advantage. Abnormal cells interior to the basal layer divide, push, and then replace the neighbouring cells leaving the overall configuration unchanged except at the border where the abnormal cells exert a thrust of $\kappa - 1$ on their normal neighbours, that is $\frac{\mathrm{d}N}{\mathrm{d}t} = (\kappa - 1)n$ where n and N are respectively the numbers of peripheral and total abnormal cells. They found that dimensionality of fractal is involved and the dimension 1.1, instead of 1, must be assigned to the periphery, which means that n is proportional to $N^{0.55}$ not $N^{0.5}$. They found that abnormal cells push out faster in this order: the triangular, square, and hexagonal lattice. We may explain this, by looking at the coordination numbers of these three lattices, that the higher coordination number the lattice sites have, the slower the cluster expands. Added coordination means a higher degree-of-freedom the newly divided cells have to move about while still remaining local.

There is no percolation in one dimension because in such case the percolating cluster must necessarily contain the whole space. But there are some applications where the critical blockage is important, for example the blockage of drainage grilles by pea shingle is one-dimensional. Here the critical amount of blockage depends on the critical rate of flow which in turn depends on the amount of water and the rate of accumulation of water to be drained. The maintenance of gullies and grilles is done by cleaning, flushing and grit bucket emptying (*cf* Harrison and Trotman, 2002).

Voronoi percolation, 2-d

The percolation of Voronoi tessellation in two dimensions can be achieved by the following algorithm.

```
generate random points;
generate Voronoi tessellation and Delaunay triangulation;
find vertices within the square box bounded by lower and upper bounds;
find neighbours of all cells, bonds, vertices, and edges;
for unit = cell, bond, vertice and edge do
  find number of units;
  permute list of units;
  clear cluster list 1, cluster list 2, set of resultant clusters;
  cluster percolated ← false;
  for $i = 1$ to number of units do
    existing cluster joined = false;
```

```
        for $j = 1$ to number of clusters in cluster list 1 do
          if the $j^{th}$ cluster contains the $i^{th}$ unit in permuted list do
            merge the $i^{th}$ unit into the $j^{th}$ cluster ;
            existing cluster joined $\leftarrow$ true;
          end
          if existing cluster joined is true
            move the $j^{th}$ cluster of cluster list 1 to cluster list 2 ;
            for $k = 1$ to number of clusters in cluster list 1 do
              if the $k^{th}$ cluster in list 1 touches the cluster in list 2 do
                merge the former into the latter;
              else
                append the former to cluster list 2;
              end
            end
            test percolation of the cluster just updated;
            move cluster list 2 to cluster list 1;
            clear cluster list 2;
            break;
          end
        end
        if existing cluster joined is false
          create a new cluster of size one and append it to list 1;
        end
        append cluster list 1 to set of resultant clusters ;
      end
    end
```

Algorithm 1 *Network percolation in 2-d.*

Pc probabilities are found to be ${}^{Vn\,2d}_{4(1),14(2)}p_c = 0.5110 \pm 0.0856$, ${}^{Vn\,2d}_{17(2)}p_b = 0.3095 \pm 0.0523$, ${}^{Vn\,2d}_{16(2)}p_v = 0.7231 \pm 0.0616$ and ${}^{Vn\,2d}_{16(2)}p_e = 0.6801 \pm 0.0468$.

And coordination numbers are $.2436$ ${}^{Vn\,2d}_{2(1),7(2)}x_c = 5.2320$, $.2979$ ${}^{Vn\,2d}_{8(2)}x_b = 9.5022$, $.0259$ ${}^{Vn\,2d}_{8(2)}x_v = 2.8617$ and $.0382$ ${}^{Vn\,2d}_{8(2)}x_e = 3.8064$.

Voronoi percolation, 3-d

In three dimensions the procedure of finding p_c is similar to that used in the 2-d case. Algorithm 2 is what I wrote and used in doing the simulation. I developed it to run on Matlab, and therefore do every thing in matrix form. As a consequence, there are various types of data all of which are matrices. These data structures can be summarised into the following.

A *list* is a one-dimensional matrix whose dimension is the number of its members. An *index* is a matrix whose members are either one or zero. These index matrices in two dimensions are like a map or an array. They are normally one or two dimensions, and map the relationship between the members of one set and another. A *set* is an $m \times n$ matrix every one of the members a_{ij} of which is a matrix. A *pair matrix* is a square diagonal index matrix that maps among members within the same set. It contains the information about relationships like neighbourhood, group membership, connection, *etc*. A pair matrix or an index can also contain information other than the existence or membership flags, for example the edge length in the case of a vertices pair edge matrix. In a bond or an edge the mid point represents its position.

The programme can be divided into three parts, creating and arranging the Voronoi data, finding the neighbourhood matrix and the percolation simulation for p_c. Putting the vertices of a face in order amounts to juggling from one end of each stick, *i.e.* edge, to another. Edges are traced in one direction only until all the edges are successfully linked head to tail. Two lists receive the result from the tracing, vertices which match go to one of them while those which do not is put in the other and recycled.

Algorithm 2 *Managing the Voronoi data in three dimensions.*

```
$x \leftarrow$ create random points;
$(v_a, c_a) \leftarrow$ create Voronoi tessellation;
$t \leftarrow$ create Delaunay tessellation;
find $v_n$ such that for all $v_a$, $0 < v_a < 1$;
find $c_n$ such that all vertices of $c_a$ are in $v_n$;
$c \leftarrow c_n$;
for all  do
  $m_b \leftarrow$ find mid bond coordinates;
  $b_l \leftarrow$ find bond length;
endfor
for all pairs of cells do
  $f_a \leftarrow$ find shared vertices when either of the two cells is in $c_n$;
endfor
for all $v_n$, $v \leftarrow v_a$;
for all $c_n$, $f \leftarrow f_a$;
for all $f$ do
  if it has three vertices then
    all the three possible pair combinations are neighbours;
  else
    $(a, b, c) \leftarrow$ find the face equation from three vertices;
```

```
        $\theta_i \leftarrow ka$ where $k = 1/(a^2 + b^2 + c^2)$ and $i = 1, 2$ and $3$;
        max $\leftarrow 0$;
        **for** $j = 1$ to $3$ **do**
          **if** $\theta <$ max **then**
            $j_m \leftarrow j$;
            max $\leftarrow \theta(j_m)$;
          **endif**
        **endfor**
        **if** $j_m = 1$ **then**
          $p \leftarrow y$; $q \leftarrow z$;
        **elseif** $j_m = 2$ **then**
          $p \leftarrow x$; $q \leftarrow z$;
        **else**
          $p \leftarrow x$; $q \leftarrow y$;
        **endif**
        $d \leftarrow$ **find** delaunay triangulation from $p$ and $q$;
        **for** all edges of all triangles of $d$ **do**
          **record** the number of times they occur;
        **endfor**
        neighbours $\leftarrow$ vertices of edges which occur only once;
        **for** all $f$ **do**
          **order** all their vertices;
        **end**
      **endif**
    **endfor**
```
□

The terms *vertices* and *edges* refer to the Voronoi tessellation only. The vertices and edges of the Delaunay tessellation are cells and bonds of the VT. The programme `perco3d.m` can find p_c for all of these. For each one of them we must find the neighbourhood matrix as well as lists of the upper- and the lower boundaries. The order of blockages is completely decided in advance by finding a permuted list of numbered items. The programme finds the history of clusters for the whole range of p, *i.e.* from 0 to 1.

To obtain the matrix of cell neighbours, first crop out between 5 and 10 per cent the boundary in all directions. Then the neighbour matrix is found from the DT matrix.

The bond neighbour matrix is simply the rearrangement of the cell neighbour matrix obtained. We have already found the vertices neighbour matrix earlier while searching for vertices shared between cells. This is rearranged for the use in the percolation programme, and then again for the edge neighbour matrix.

Next is Algorithm 3 which carries out the percolation simulation. The cluster information is swapped alternately between three variables,

A, B and tmp, to facilitate the flow; $A(i)$ is the i^{th} cluster in A. All variables are assumed to be cleared at the start of the algorithm. At the end of each run the value of p_c is computed, and the list p is reversed and reused for a second time if it was the first run.

Algorithm 3 *Voronoi percolation in three dimensions.*

```
for each permuted item $p(i)$ in the list do
  joined ← 0;
  for each cluster $A(j)$ in $A$ do
    if A(j) contains p(i) then
      $A(j) \leftarrow A(j) \cap p(i)$;
      $B(1) \leftarrow A(j)$;
      $tmp \leftarrow A$;
      $A \leftarrow tmp - A(j)$;
      for all $A(k)$ in $A$ do
        if $A(k) \wedge B(1)$ then
          $B(1) \leftarrow A(k) \cap B(1)$;
        else
          $B(++n_B) \leftarrow A(k)$;
        endif
      endfor
      percolated ?;
      $A \leftarrow B$;
      break;
    endif
  endfor
  if joined then
    $A(++n_A) \leftarrow p(i)$;
  endif
endfor                                              □
```

The resulted Pc probabilities from simulation are ${}^{Vn\,3d}_{12(2)}p_c = 0.2340 \pm 0.0448$, ${}^{Vn\,3d}_{2(2),10(3)}p_b = 0.1178 \pm 0.0271$, ${}^{Vn\,3d}_{2(2),10(3)}p_v = 0.2941 \pm 0.0831$ and ${}^{Vn\,3d}_{12(3)}p_e = 0.4311 \pm 0.0324$.

And the coordination numbers are $.3329\ {}^{Vn\,3d}_{6(2)}x_c = 11.1231$, $.5996\ {}^{Vn\,3d}_{1(2),5(3)}x_b = 23.0548$, $.0245\ {}^{Vn\,3d}_{1(2),5(3)}x_v = 3.6926$ and $.0432\ {}^{Vn\,3d}_{6(3)}x_e = 5.5002$.

Percolation, 2–d Voronoi section

The programme used does a 2-d section of the 3-d VT. Algorithm 4 describes how it works. It assumes that the Voronoi tessellation already exists. Here d_m means a denominator while n_r a numerator, V and C

means vertices and cells of the 3-d Voronoi tessellation while v and c those of the 2-d section. In particular, $c \in C$

Algorithm 4 *Plane section of Voronoi in three dimensions*

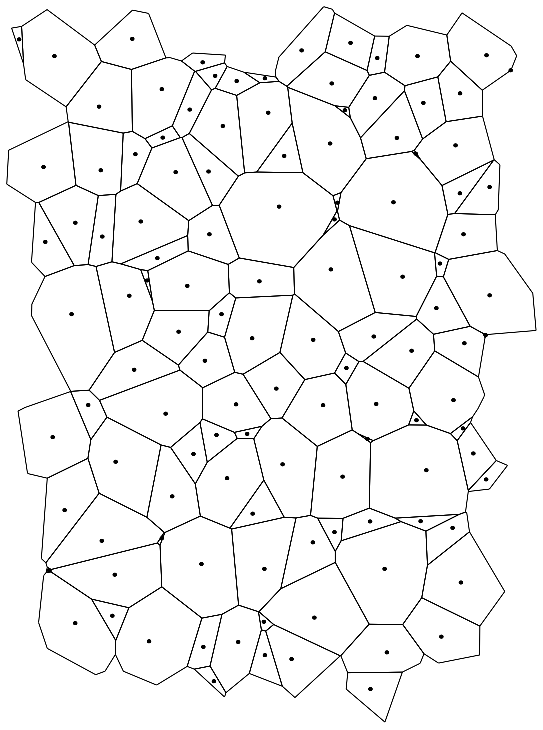
```
$(\Delta x, \Delta y, \Delta z)_i \leftarrow (x_2 - x_1, y_2 - y_1, z_2 - z_1)_i$ for all edges $i$;
for all edges $i$ do
  $d_m \leftarrow a\Delta x + b\Delta y + c\Delta z$;
  if $d_m$ nonzero then
    $n_r \leftarrow ax_1 + by_1 + cz_1$;
    $t \leftarrow -n_r/d_m$;
    if $0 \leq t \leq 1$ then
      $(x, y, z) \leftarrow (x_1 + \Delta x, y_1 + \Delta y, z_1 + \Delta z)_i$;
      $\{v_s\} \leftarrow (x, y, z)$;
    endif
  else
    $a \leftarrow a + \epsilon$;
    $d_m \leftarrow a\Delta x + b\Delta y + c\Delta z$;
    $n_r \leftarrow ax_1 + by_1 + cz_1$;
    $t \leftarrow -n_r/d_m$;
    if $0 \leq t \leq 1$ then
      $(x, y, z) \leftarrow (x_1 + \Delta x, y_1 + \Delta y, z_1 + \Delta z)_i$;
      $\{v_s\} \leftarrow (x, y, z)$;
    endif
    if $n_r = 0$ then
      $\{v_s\} \leftarrow (x_1, y_1, z_1)$;
      $\{v_s\} \leftarrow (x_2, y_2, z_2)$;
    endif
  endif
endfor
for all $c_s$ do
  find the Delaunay triangulation;
  count $n_e$ of all the triangles, add the numbers into a single list;
endfor                                                              □
```

The intersection of a plane $ax + by + cz + d = 0$ by the line which is defined by $x = x_1 + \Delta xt$, $y = y_1 + \Delta yt$ and $z = z_1 + \Delta zt$, where Δx, Δy and Δz are respectively $(x_2 - x_1)$, $(y_2 - y_1)$ and $(z_2 - z_1)$, is determined by $t = -(ax_1 + by_1 + cz_1 + d)/(ai\Delta x + b\Delta y + c\Delta z)$. If the denominator is zero the line is parallel to the plane, and if the nominator is also zero contained therein.

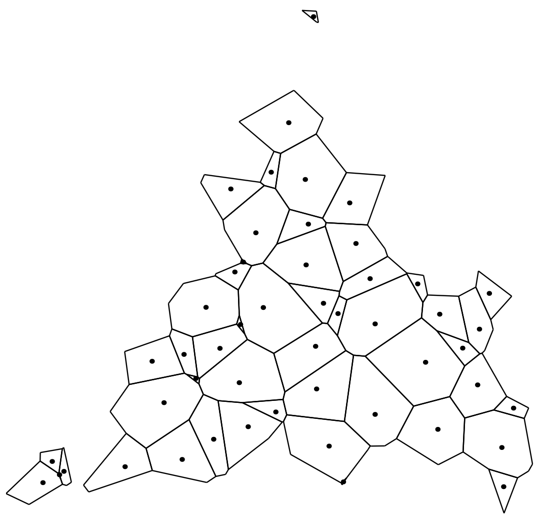
The results from the percolation simulation on the section $\mathcal{V}_3^2$ are ${}^{Vn\,(2,3)s}_{18(1),4(2)}p_c = 0.5494 \pm 0.1223$, ${}^{Vn\,(2,3)s}_{10(1),10(2)}p_b = 0.3515 \pm 0.0764$, ${}^{Vn\,(2,3)s}_{8(1),12(2)}p_v = 0.7557 \pm 0.0757$, ${}^{Vn\,(2,3)s}_{20(2)}p_e = 0.6210 \pm 0.0665$.

The coordination numbers are .2212 ${}^{Vn\,(2,3)s}_{8(\mathbf{1}),1(\mathbf{2})}x_c = 4.6894$, .6021 ${}^{Vn\,(2,3)s}_{5(\mathbf{1}),4(\mathbf{2})}x_b = 8.8100$, .0387 ${}^{Vn\,(2,3)s}_{4(\mathbf{1}),5(\mathbf{2})}x_v = 2.7495$, .1118 ${}^{Vn\,(2,3)s}_{9(\mathbf{2})}x_e = 3.6691$.

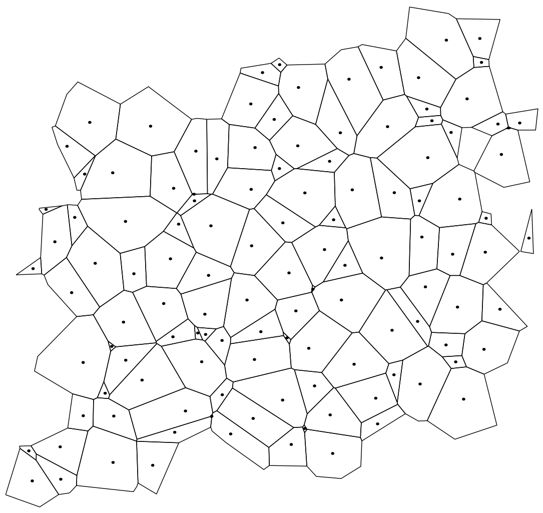
When sectioned by the plane $(a, b, c, d) = (0.01,\ 0.5,\ 0.5,\ -0.5)$, our VT gives a picture as shown in Figure 42. Here the points shown are merely the average of the coordinates of all vertices in each cell. From ten simulations of these 118 cells, 300 bonds, 288 vertices and 405 edges having respectively the coordinate numbers of 5.0847, 9.7933, 2.8125 and 3.7333, we obtain $p_c = 0.5220 \pm 0.0966$, $p_b = 0.3107 \pm 0.0391$, $p_v = 0.7014 \pm 0.0573$ and $p_e = 0.6259 \pm 0.0603$.



When sectioned by the plane $(a, b, c, d) = (0.5,\ -0.5,\ 0.5,\ 0.01)$, our VT gives a picture as shown in Figure 43. Here the points shown are merely the average of the coordinates of all vertices in each cell. From fourteen simulations of these 50 cells, 104 bonds, 140 vertices and 187 edges having respectively the coordinate numbers of 4.1600, 8.2500, 2.6714 and 3.5080, we obtain $p_c = 0.6914 \pm 0.1424$, $p_b = 0.4533 \pm 0.1108$, $p_v = 0.7760 \pm 0.0821$ and $p_e = 0.6929 \pm 0.0897$.:



When sectioned by the plane $(a, b, c, d) = (-0.7, -0.3, 1, 0.01)$, our VT gives a picture as shown in Figure 44. Here the points shown are merely the average of the coordinates of all vertices in each cell. From twenty simulations of these 121 cells, 305 bonds, 291 vertices and 411 edges having respectively the coordinate numbers of 5.0413, 9.4426, 2.8247 and 3.7518, we obtain $p_c = 0.5413 \pm 0.1107$, $p_b = 0.3584 \pm 0.0494$, $p_v = 0.7077 \pm 0.0572$ and $p_e = 0.6749 \pm 0.0470$.:



For the purpose of these simulations, I have turned the code for finding percolation into a function. To my surprise and delight, this same function works for both the 3–d VT and its 2–d section. In choosing a sectioning plane it is better if we choose the parameter d small, as the plane will then pass close to the origin. Also, choosing $a+b+c \approx 0$ seems to make a more wholesome section than otherwise. The codes for sectioning work well for oblique planes but do not like planes which are parallel to an axis. This shortcoming can be avoided if we make our plane only nearly parallel, when we want it to be parallel to an axis. Then to be able to view in a head on fashion such planes which have been plotted in three dimensions, we can look from the position (a, b, c), which is in effect the vector normal to the plane. The function mentioned above is listed as `perc.m`.

Network percolation

In the study of networks an important parameter is the coordination number, which is the number of neighbours of an element which in the graph theory is usually the vertex. Each vertex or site of a graph is connected to each of its neighbouring vertices by a bond, so the coordination number of a graph is the number of bonds connected to a vertex.

Clusters and their various characteristics play an important role in the study of percolation of networks. With a material science application in mind, Levy *et al* (1982) numerically represent the shape of a cluster by the *shape parameter* S, defined as $S = b/N$ or more generally $S = (1/2)\sum_{i=1}^{z} i\nu_i / \sum_{i=1}^{z} \nu_i$ where b or i is the number of bonds and N or ν_i the number of elements in a cluster having b or i bonds respectively.

Programmes

Network percolation, two dimensions

```
% perco
clear all; St=sum(100*clock); rand('state',St); CNa=200;
Dim=2; X=rand(CNa,Dim); [Va,Ca]=voronoin(X); T=delaunayn(X);
TN=size(T,1); VNa=size(Va,1); LB=0.05; UB=0.95;
IXa=zeros(VNa,1);  V=[]; Count=0; VCNa=[];
for i=1:CNa,
  VCNa=[VCNa;size(Ca{i},2)];
end
for i=1:VNa,
  if((Va(i,1)>LB & Va(i,1)<UB) & (Va(i,2)>LB & Va(i,2)<UB))
    V=[V;Va(i,:)]; Count=Count+1; IXa(i,1)=Count;
  end
end
VN=size(V,1); VCN=[]; Count=0; Xa=X; X=[];
Tmp=sparse(1,CNa);
for i=1:CNa,
  Include=1;
  for j=1:VCNa(i,1),
    if(IXa(Ca{i}(1,j),1)==0)
```

```
      Include=0;
    end
  end
  if(Include==1)
    Count=Count+1; C{Count,1}=[]; VCN=[VCN;VCNa(i,1)];
    for j=1:VCNa(i,1),
      C{Count,1}(1,j)=IXa(Ca{i}(1,j),1);
    end
    X=[X;Xa(i,:)]; Tmp(1,i)=Count;
  end
end
CN=size(C,1); T2=[]; T3=[];
for i=1:TN,
  TmpA=[];
  for j=1:3,
    if(Tmp(T(i,j)))
      TmpA=[TmpA,Tmp(T(i,j))];
    end
  end
  TmpB=size(TmpA,2);
  if(TmpB==2)
    T2=[T2;TmpA];
  elseif(TmpB==3)
    T3=[T3;TmpA];
  end
end
% for cells
B=[]; BXX=sparse(CN,CN); NeCMat=sparse(CN,CN); Count=0;
for i=1:size(T2,1),
  Count=Count+1; B=[B;[T2(i,1),T2(i,2)]];
  BXX(T2(i,1),T2(i,2))=Count;
  BXX(T2(i,2),T2(i,1))=Count; NeCMat(T2(i,1),T2(i,2))=1;
  NeCMat(T2(i,2),T2(i,1))=1;
end
for i=1:size(T3,1),
  for j=1:Dim,
    for k=(j+1):(Dim+1),
      if(BXX(T3(i,j),T3(i,k))==0)
        Count=Count+1; B=[B;[T3(i,j),T3(i,k)]];
        BXX(T3(i,j),T3(i,k))=Count; BXX(T3(i,k),T3(i,j))=Count;
        NeCMat(T3(i,j),T3(i,k))=1; NeCMat(T3(i,k),T3(i,j))=1;
      end
    end
  end
end
BN=Count; A=X; N=size(A,1); LMat=sparse(1,N);
UMat=sparse(1,N); LBc=0.2; UBc=1-LBc;
for i=1:N,
  if(A(i,1)<=LBc)
    LMat(1,i)=1;
  elseif(A(i,1)>=UBc)
    UMat(1,i)=1;
  end
end
NeMat=NeCMat; Blocked=randperm(CN);
% for bonds
NeBMat=sparse(BN,BN);
for i=1:CN,
  [a,b,c]=find(BXX(i,:)); nc=size(c,2);
  for j=1:(nc-1),
    for k=(j+1):nc,
      NeBMat(c(1,j),c(1,k))=1; NeBMat(c(1,k),c(1,j))=1;
    end
  end
end
A=B; N=size(A,1); LMat=sparse(1,N); UMat=sparse(1,N);
for i=1:N,
  if((X(A(i,1),1)<=LBc) | (X(A(i,2),1)<=LBc))
    LMat(1,i)=1;
  elseif((X(A(i,1),1)>=UBc) | (X(A(i,2),1)>=UBc))
    UMat(1,i)=1;
  end
end
NeMat=NeBMat; Blocked=randperm(BN);
%for vertices
Tmp=sparse(1,VN);
for i=1:CN,
  for j=1:VCN(i,1),
    Tmp(1,C{i}(1,j))=1;
  end
end
Vv=[];
Count=0;
for i=1:VN,
  if(Tmp(1,i))
    Count=Count+1; Vv=[Vv;V(i,:)]; Tmp(1,i)=Count;
  end
end
VvN=size(Vv,1);
for i=1:CN,
  for j=1:VCN(i,1),
    Cv{i}(1,j)=Tmp(1,C{i}(1,j));
  end
end
E=[]; EVV=sparse(VN,VN); EVVv=sparse(VvN,VvN);
NeVMat=sparse(VvN,VvN); Countv=0; Count=0;
for i=1:CN,
  Tmp=[Cv{i}(1,1:VCN(i,1)),Cv{i}(1,1)];
  for j=1:VCN(i,1),
    V1=Tmp(1,j); V2=Tmp(1,(j+1));
    if(NeVMat(V1,V2)==0)
      Countv=Countv+1; NeVMat(V1,V2)=1; NeVMat(V2,V1)=1;
    end
  end
  Tmp=[C{i}(1,1:VCN(i,1)),C{i}(1,1)];
  for j=1:VCN(i,1),
    V1=Tmp(1,j); V2=Tmp(1,(j+1));
    if(EVV(V1,V2)==0)
      Count=Count+1; E=[E;[V1,V2]];
      EVV(V1,V2)=Count; EVV(V2,V1)=Count;
    end
  end
end
EN=Count; A=Vv; N=size(A,1); LMat=sparse(1,N);
UMat=sparse(1,N); LBv=2*LB; UBv=(UB-LB);
for i=1:N,
  if(A(i,1)<=LBv)
    LMat(1,i)=1;
  elseif(A(i,1)>=UBv)
    UMat(1,i)=1;
  end
end
NeMat=NeVMat; Blocked=randperm(VvN);
%for edges
NeEMat=sparse(EN,EN); MEV=sparse(EN,VN);
[a,b,c]=find(EVV); nc=size(c,1);
```

```
for i=1:nc,
  MEV(c(i),a(i))=1; MEV(c(i),b(i))=1;
end
for i=1:VN,
  a=find(MEV(:,i));
  if(~isempty(a))
    TmpN=size(a,1); Tmp=[a;a(1,1)]';
    for j=1:TmpN,
      for k=(j+1):TmpN,
        NeEMat(Tmp(1,j),Tmp(1,k))=1;
        NeEMat(Tmp(1,k),Tmp(1,j))=1;
      end
    end
  end
end
A=E; N=size(A,1); LMat=sparse(1,N); UMat=sparse(1,N);
for i=1:N,
  if((V(A(i,1),1)<=LBv) | (V(A(i,2),1)<=LBv))
    LMat(1,i)=1;
  elseif((V(A(i,1),1)>=UBv) | (V(A(i,2),1)>=UBv))
    UMat(1,i)=1;
  end
end
NeMat=NeEMat; Blocked=randperm(EN);
% perco1
clear ClusA ClusB TSeries; NClusA=0; Perco=0;
for i=1:N,
  Joined=0;
  for j=1:NClusA,
    if(ClusA{j,3}(1,Blocked(1,i))~=0)
      ClusA{j,1}=ClusA{j,1}+1; ClusA{j,2}(1,Blocked(1,i))=1;
      ClusA{j,3}=ClusA{j,3} | NeMat(Blocked(1,i),:); Joined=1;
    end
    if(Joined==1)
      for k=1:4,
        ClusB{1,k}=ClusA{j,k};
      end
      NClusB=1;
      if(j==1)
        Tmp=ClusA; clear ClusA;
        for k=1:(NClusA-1),
          for l=1:4,
            ClusA{k,l}=Tmp{(k+1),l};
          end
        end
      elseif(j==NClusA)
        Tmp=ClusA; clear ClusA;
        for k=1:(NClusA-1),
          for l=1:4,
            ClusA{k,l}=Tmp{k,l};
          end
        end
      else
        Tmp=ClusA; clear ClusA;
        for k=1:(j-1),
          for l=1:4,
            ClusA{k,l}=Tmp{k,l};
          end
        end
        for k=j:(NClusA-1),
          for l=1:4,
            ClusA{k,l}=Tmp{(k+1),l};
          end
        end
      end
      for k=1:(NClusA-1),
        if(sum(ClusA{k,2} & ClusB{1,3}) ~= 0)
          ClusB{1,1}=ClusB{1,1}+ClusA{k,1};
          ClusB{1,2}=ClusB{1,2} | ClusA{k,2};
          ClusB{1,3}=ClusB{1,3} | ClusA{k,3};
          ClusB{1,4}=ClusB{1,4} | ClusA{k,4};
        else
          NClusB=NClusB+1;
          for l=1:4,
            ClusB{NClusB,l}=ClusA{k,l};
          end
        end
      end
      if((sum(full(LMat & ClusB{1,2}))~=0) & ...
         (sum(full(UMat & ClusB{1,2}))~=0))
        ClusB{1,4}=1; Perco=1;
      end
      NClusA=NClusB; ClusA=ClusB; clear ClusB; break;
    end
  end
  if(Joined==0)
    NClusA=NClusA+1; ClusA{NClusA,1}=1;
    ClusA{NClusA,2}=sparse(1,Blocked(1,i),1,1,N);
    ClusA{NClusA,3}=NeMat(Blocked(1,i),:); ClusA{NClusA,4}=0;
  end
  TSeries{i,1}=ClusA; TSeries{i,2}=Perco;
end
% Reverse
Tmp=Blocked; Blocked=[];
for i=1:N,
  Blocked=[Blocked,Tmp(1,(N-i+1))];
end
% simulations
Nc=0; TSnap=[];
for i=1:N,
  if(TSeries{i,2})
    Nc=i; break;
  end
end
Pc=Nc/N; Cord=mean(sum(NeMat,2));
```

Network percolation, three dimensions

```
% perco3d.m
clear all; St=sum(100*clock); rand('state',St);
CNa=300; Dim=3; X=rand(CNa,Dim); [Va,Ca]=voronoin(X);
T=delaunayn(X); TN=size(T,1); VNa=size(Va,1); LB=0.05;
UB=0.95; IXa=zeros(VNa,1); VCNa=[];
for i=1:CNa,
  VCNa=[VCNa;size(Ca{i},2)];
end
MVCa=[];
for i=1:CNa,
  Tmp=ones(1,VCNa(i,1));
  MVCa=[MVCa;sparse(Tmp,Ca{i},Tmp,1,VNa)];
end
Vin=zeros(1,VNa); Count=0;
for i=1:VNa,
  if((max(Va(i,:))<1) & (min(Va(i,:))>0))
    Count=Count+1; Vin(1,i)=1; IXa(i,1)=Count;
  end
end
Tmp=~Vin; Cin=ones(1,CNa);
for i=1:CNa,
  if(sum(Tmp & MVCa(i,:)))
    Cin(1,i)=0;
  end
end
C=[]; count=0; VCN=[];
for i=1:CNa,
  if(Cin(i))
    count=count+1; TmpN=size(Ca{i},2);
    for j=1:TmpN,
      C{count,1}(1,j)=IXa(Ca{i}(1,j));
    end
    VCN(count,1)=TmpN;
  end
end
CN=size(C,1); MidBCx=sparse(CNa,CNa); MidBCy=sparse(CNa,CNa);
MidBCz=sparse(CNa,CNa); BLng=sparse(CNa,CNa);
for i=1:TN,
  Tmp=[T(i,:),T(i,1)];
  for j=1:(Dim+1),
    for k=(j+1):(Dim+1),
      if((Cin(1,Tmp(1,j)) | Cin(1,Tmp(1,k))) & ~BLng(j,k))
        MidBCx(Tmp(1,j),Tmp(1,k))=(X(k,1)+X(j,1))/2;
        MidBCx(Tmp(1,k),Tmp(1,j))=(X(k,1)+X(j,1))/2;
        MidBCy(Tmp(1,j),Tmp(1,k))=(X(k,2)+X(j,2))/2;
        MidBCy(Tmp(1,k),Tmp(1,j))=(X(k,2)+X(j,2))/2;
        MidBCz(Tmp(1,j),Tmp(1,k))=(X(k,3)+X(j,3))/2;
        MidBCz(Tmp(1,k),Tmp(1,j))=(X(k,3)+X(j,3))/2;
        dx=X(k,1)-X(j,1); dy=X(k,2)-X(j,2); dz=X(k,3)-X(j,3);
        TmpA=sqrt(dx*dx + dy*dy + dz*dz);
        BLng(Tmp(1,j),Tmp(1,k))=TmpA;
        BLng(Tmp(1,k),Tmp(1,j))=TmpA;
      end
    end
  end
end
Fa=[]; Count=0; FaC=[];
for i=1:CNa,
  if(Cin(1,i))
    FaC{i,1}=0; FaC{i,2}=[];
  end
end
for i=1:(CNa-1),
  for j=(i+1):CNa,
    TmpA=0; TmpB=0;
    if(Cin(1,i))
      TmpA=1;
    end
    if(Cin(1,j))
      TmpB=1;
    end
    if(TmpA | TmpB)
      Tmp=MVCa(i,:) & MVCa(j,:);
      if(sum(Tmp))
        [a,b]=find(Tmp); Count=Count+1;
        Fa{Count,1}=size(b,2); Fa{Count,2}=b;
        Fa{Count,3}=[MidBCx(i,j),MidBCy(i,j),MidBCz(i,j)];
        if(TmpA)
          FaC{i,1}=FaC{i,1} + 1; FaC{i,2}=[FaC{i,2},Count];
          FaC{i,3}{1,1}=i; FaC{i,3}{1,2}=j;
          end
          if(TmpB)
            FaC{j,1}=FaC{j,1} + 1; FaC{j,2}=[FaC{j,2},Count];
            FaC{j,3}{1,1}=i; FaC{j,3}{1,2}=j;
          end
        end
      end
    end
  end
FaN=size(Fa,1); V=[];
for i=1:VNa,
  if(Vin(1,i))
    V=[V;[Va(i,:),i]];
  end
end
VN=size(V,1); Tmp=sparse(VNa,1);
for i=1:VN,
  Tmp(V(i,4),1)=i;
end
F=Fa;
for i=1:FaN,
  for j=1:F{i,1},
    F{i,2}(1,j)=Tmp(Fa{i,2}(1,j),1);
  end
end
FN=FaN; FC=[];
count=0;
for i=1:CNa,
  if(Cin(i))
    count=count+1; FC{count,1}=FaC{i,1};
    FC{count,2}=FaC{i,2}; FC{count,3}=FaC{i,3};
  end
end
NghV=sparse(VN,VN); Tmp=F; TmpN=FN;
for i=1:TmpN,
  TmpA=Tmp{i,2}; x=[]; y=[]; z=[]; TmpB=Tmp{i,1};
  if(TmpB==3)
    for j=1:2,
      for k=(j+1):3,
        NghV(TmpA(1,j),TmpA(1,k))=1;
        NghV(TmpA(1,k),TmpA(1,j))=1;
      end
    end
  else
    for j=1:TmpB,
      x=[x;V(TmpA(1,j),1)];
      y=[y;V(TmpA(1,j),2)];
      z=[z;V(TmpA(1,j),3)];
    end
    a=y(1)*(z(2)-z(3))+y(2)*(z(3)-z(1))+y(3)*(z(1)-z(2));
    b=z(1)*(x(2)-x(3))+z(2)*(x(3)-x(1))+z(3)*(x(1)-x(2));
    c=x(1)*(y(2)-y(3))+x(2)*(y(3)-y(1))+x(3)*(y(1)-y(2));
    K=1/sqrt(a*a + b*b + c*c);
    Th{1}=K*a; Th{2}=K*b; Th{3}=K*c; Max=0;
    for j=1:3,
      if(Th{j}<Max)
        Max=Th{j}; jMax=j;
      end
    end
    if(jMax==1)
      p=y; q=z;
    elseif(jMax==2)
      p=x; q=z;
    else
      p=x; q=y;
    end
    t=delaunay(p,q);
    for j=1:size(t,1),
      for k=1:3,
        t(j,k)=TmpA(1,t(j,k));
      end
    end
    Nt=size(t,1); TmpC=sparse(VN,VN);
    for j=1:Nt,
      TmpT=[t(j,:),t(j,1)];
      for k=1:3,
        TmpD=sort([TmpT(1,k),TmpT(1,(k+1))]);
        TmpC(TmpD(1,1),TmpD(1,2))=TmpC(TmpD(1,1),TmpD(1,2))+1;
      end
    end
    [k,l,m]=find(TmpC);
    for j=1:size(k,1),
      if(m(j)==1)
        NghV(k(j),l(j))=1; NghV(l(j),k(j))=1;
      end
    end
  end
end
Fed=[];
for i=1:FN,
  for j=1:2,
    Fed{i,j}=F{i,j};
  end
end
for i=1:FN,
  Count=0; TmpN=Fed{i,1};
  if(TmpN>3)
    Tmp=Fed{i,2}; TmpA=Tmp(1,1); Tmp=Tmp(1,2:TmpN);
    TmpN=TmpN-1; Count=Count+1;
    while(TmpN)
      a=TmpA(1,Count); TmpB=[]; Found=0;
      for j=1:TmpN,
        TmpC=Tmp(1,j);
        if(NghV(a,TmpC) & ~Found)
          TmpA=[TmpA,TmpC]; Found=1;
        else
          TmpB=[TmpB,TmpC];
        end
      end
      Tmp=TmpB; TmpN=TmpN-1; Count=Count+1;
    end
    Fed{i,2}=TmpA;
  end
end
LB2=2*LB; UB2=(UB-LB);
% cells II
Tmp=ones(1,CNa);
for i=1:CNa,
  if((max(X(i,:))>UB) | (min(X(i,:))<LB))
    Tmp(1,i)=0;
  end
end
a=find(Tmp); TmpB=sparse(1,CNa); x=[];
for i=1:size(a,2),
  TmpB(1,a(1,i))=i; x(i,:)=X(a(i),:);
end
```

```
xn=size(x,1); TmpA=zeros(size(T)); TmpN=size(T,1);
for i=1:TmpN,
  for j=1:4,
    if((Tmp(1,T(i,j))))
      TmpA(i,j)=TmpB(1,T(i,j));
    end
  end
end
nghc=sparse(xn,xn);
for i=1:TmpN,
  [a,b,c]=find(TmpA(i,:)); TmpB=size(c,2);
  if(TmpB>1)
    for j=1:(TmpB-1),
      for k=(j+1):TmpB,
        nghc(c(1,j),c(1,k))=1; nghc(c(1,k),c(1,j))=1;
      end
    end
  end
end
A=x; N=size(A,1); LMat=sparse(1,N); UMat=sparse(1,N);
for i=1:N,
  if(A(i,1)<=LB2)
    LMat(1,i)=1;
  elseif(A(i,1)>=UB2)
    UMat(1,i)=1;
  end
end
NeMat=nghc; Blocked=randperm(xn);
% bonds II
[a,b,c]=find(triu(nghc));
b=[a,b]; bn=size(b,1); Tmp=sparse(bn,xn);
for i=1:bn,
  Tmp(i,b(i,1))=1; Tmp(i,b(i,2))=1;
end
nghb=sparse(bn,bn);
for i=1:xn,
  a=find(Tmp(:,i));
  if(~isempty(a))
    TmpN=size(a,1);
    for j=1:(TmpN-1),
      for k=(j+1):TmpN,
        nghb(a(j,1),a(k,1))=1; nghb(a(k,1),a(j,1))=1;
      end
    end
  end
end
A=b; N=size(A,1); LMat=sparse(1,N); UMat=sparse(1,N);
for i=1:N,
  if((x(A(i,1),1)<=LB2) | (x(A(i,2),1)<=LB2))
    LMat(1,i)=1;
  elseif((x(A(i,1),1)>=UB2) | (x(A(i,2),1)>=UB2))
    UMat(1,i)=1;
  end
end
NeMat=nghb; Blocked=randperm(N);
% vertices II
[a,b,c]=find((NghV)); Tmp=sparse(1,VN);
for i=1:size(a,1),
  Tmp(1,b(i,1))=1;
end
d=find(Tmp); TmpN=size(d,2);
for i=1:TmpN,
  Tmp(1,d(1,i))=i;
```

```
end
for i=1:size(a,1),
  a(i,1)=Tmp(1,a(i,1)); b(i,1)=Tmp(1,b(i,1));
end
nghv=sparse(a,b,c,TmpN,TmpN); TmpA=[];
for i=1:TmpN,
  TmpA(Tmp(1,d(i)),:)=V(i,1:3);
end
A=TmpA; N=size(A,1); LMat=sparse(1,N); UMat=sparse(1,N);
for i=1:N,
  if(A(i,1)<=LB2)
    LMat(1,i)=1;
  elseif(A(i,1)>=UB2)
    UMat(1,i)=1;
  end
end
NeMat=nghv; Blocked=randperm(N);
% edges II
[a,b,c]=find(triu(NghV)); E=[a,b]; EN=size(E,1);
Tmp=sparse(EN,VN);
for i=1:EN,
  Tmp(i,E(i,1))=1; Tmp(i,E(i,2))=1;
end
NghE=sparse(EN,EN);
for i=1:VN,
  a=find(Tmp(:,i));
  if(~isempty(a))
    TmpN=size(a,1);
    for j=1:(TmpN-1),
      for k=(j+1):TmpN,
        NghE(a(j,1),a(k,1))=1; NghE(a(k,1),a(j,1))=1;
      end
    end
  end
end
A=E; N=size(A,1); LMat=sparse(1,N); UMat=sparse(1,N);
for i=1:N,
  if((V(A(i,1),1)<=LB2) | (V(A(i,2),1)<=LB2))
    LMat(1,i)=1;
  elseif((V(A(i,1),1)>=UB2) | (V(A(i,2),1)>=UB2))
    UMat(1,i)=1;
  end
end
NeMat=NghE; Blocked=randperm(N);
% percolation
clear ClusA ClusB TSeries; NClusA=0; Perco=0;
for i=1:N,
  Joined=0;
  for j=1:NClusA,
    if(ClusA{j,3}(1,Blocked(1,i))~=0)
      ClusA{j,1}=ClusA{j,1}+1;
      ClusA{j,2}(1,Blocked(1,i))=1;
      ClusA{j,3}=ClusA{j,3} | NeMat(Blocked(1,i),:);
      Joined=1;
    end
    if(Joined==1)
      for k=1:4,
        ClusB{1,k}=ClusA{j,k};
      end
      NClusB=1;
      if(j==1)
        Tmp=ClusA; clear ClusA;
        for k=1:(NClusA-1),
          for l=1:4,
            ClusA{k,l}=Tmp{(k+1),l};
          end
        end
      elseif(j==NClusA)
        Tmp=ClusA; clear ClusA;
        for k=1:(NClusA-1),
          for l=1:4,
            ClusA{k,l}=Tmp{k,l};
          end
        end
      else
        Tmp=ClusA; clear ClusA;
        for k=1:(j-1),
          for l=1:4,
            ClusA{k,l}=Tmp{k,l};
          end
        end
        for k=j:(NClusA-1),
          for l=1:4,
            ClusA{k,l}=Tmp{(k+1),l};
          end
        end
      end
      for k=1:(NClusA-1),
        if(sum(ClusA{k,2} & ClusB{1,3}) ~= 0)
          ClusB{1,1}=ClusB{1,1}+ClusA{k,1};
          ClusB{1,2}=ClusB{1,2} | ClusA{k,2};
          ClusB{1,3}=ClusB{1,3} | ClusA{k,3};
          ClusB{1,4}=ClusB{1,4} | ClusA{k,4};
        else
          NClusB=NClusB+1;
          for l=1:4,
            ClusB{NClusB,l}=ClusA{k,l};
          end
        end
      end
      if((sum(full(LMat & ClusB{1,2}))~=0) & ...
         (sum(full(UMat & ClusB{1,2}))~=0))
        ClusB{1,4}=1; Perco=1;
      end
      NClusA=NClusB; ClusA=ClusB; clear ClusB; break;
    end
  end
  if(Joined==0)
    NClusA=NClusA+1; ClusA{NClusA,1}=1;
    ClusA{NClusA,2}=sparse(1,Blocked(1,i),1,1,N);
    ClusA{NClusA,3}=NeMat(Blocked(1,i),:);
    ClusA{NClusA,4}=0;
  end
  TSeries{i,1}=ClusA; TSeries{i,2}=Perco;
end
% Reverse
Tmp=Blocked; Blocked=[];
for i=1:N,
  Blocked=[Blocked,Tmp(1,(N-i+1))];
end
% simulations
Nc=0;
for i=1:N,
  if(TSeries{i,2})
    Nc=i; break;
  end
```

```
end
Pc=Nc/N; Cord=mean(sum(NeMat,2));
```

Network percolation, 2–d section

```
% section
MVC=[];
for i=1:CN,
  Tmp=ones(1,VCN(i,1)); MVC=[MVC;sparse(Tmp,C{i},Tmp,1,VN)];
end
CE=[];
for i=1:EN,
  Tmp=MVC(:,E(i,1)) & MVC(:,E(i,2));
  if(sum(Tmp))
    TmpA=find(Tmp)'; TmpN=size(TmpA,2);
    CE{i,1}=TmpN; CE{i,2}=TmpA;
  end
end
ie=[]; je=[]; ke=[];
for i=1:EN,
  ie(i,1)=V(E(i,2),1)-V(E(i,1),1);
  je(i,1)=V(E(i,2),2)-V(E(i,1),2);
  ke(i,1)=V(E(i,2),3)-V(E(i,1),3);
end
a=1; b=.01; cc=0; d=-.5; v=[]; vC=[]; count=0;
for i=1:EN,
  Tmp=(a*ie(i)+b*je(i)+cc*ke(i));
  if(Tmp)
    v1=E(i,1); x1=V(v1,1); y1=V(v1,2); z1=V(v1,3);
    TmpA=(a*x1+b*y1+cc*z1+d);
    v2=E(i,2); x2=V(v2,1); y2=V(v2,2); z2=V(v2,3);
    t=-TmpA/Tmp;
    if((t>=0) & (t<=1))
      x=x1+(x2-x1)*t; y=y1+(y2-y1)*t; z=z1+(z2-z1)*t;
      count=count+1;
      v(count,:)=[x,y,z];
      vC{count,1}=CE{i,1}; vC{count,2}=CE{i,2};
    end
  else
    if(~TmpA)  % both nom and denom = 0
      count=count+1;
      v(count,:)=[x1,y1,z1]; vC{count,1}=CE{i,1};
      vC{count,2}=CE{i,2};
      count=count+1;
      v(count,:)=[x2,y2,z2];
      vC{count,1}=CE{i,1}; vC{count,2}=CE{i,2};
    end
  end
end
vn=count; cC=sparse(CN,1); count=0;
for i=1:vn,
  for j=1:vC{i,1},
    if(~cC(vC{i,2}(j)))
      count=count+1; cC(vC{i,2}(j),1)=count;
    end
  end
end
cn=count; vc=vC;
for i=1:vn,
  for j=1:vc{i,1},
    vc{i,2}(1,j)=cC(vC{i,2}(j));
  end
```

```
end
c=[];
for i=1:cn,
  c{i,1}=0; c{i,2}=[];
end
for i=1:vn,
  for j=1:vc{i,1},
    c{vc{i,2}(j),1}=c{vc{i,2}(j),1}+1;
    c{vc{i,2}(j),2}=[c{vc{i,2}(j),2},i];
  end
end
for i=1:cn,
  Tmp=[];
  for j=1:c{i,1},
    Tmp=[Tmp;v(c{i,2}(j),:),c{i,2}(j)];
  end
  TmpA=min(Tmp,[],1); TmpB=max(Tmp,[],1);
  [TmpC,TmpD]=min(TmpB-TmpA);
  if(TmpD==1)
    TmpA=Tmp(:,2); TmpB=Tmp(:,3);
  elseif(TmpD==2)
    TmpA=Tmp(:,1); TmpB=Tmp(:,3);
  else
    TmpA=Tmp(:,1); TmpB=Tmp(:,2);
  end
  TmpC=delaunay(TmpA,TmpB); TmpN=size(TmpC,1);
  for j=1:TmpN,
    for k=1:3,
      TmpC(j,k)=Tmp(TmpC(j,k),4);
    end
  end
  TmpA=sparse(vn,vn);
  for j=1:TmpN,
    for k=1:2,
      for m=(k+1):3,
        TmpA(TmpC(j,k),TmpC(j,m))=TmpA(TmpC(j,k),TmpC(j,m))+1;
        TmpA(TmpC(j,m),TmpC(j,k))=TmpA(TmpC(j,m),TmpC(j,k))+1;
      end
    end
  end
  [x,y,z]=find(TmpA); TmpB=[]; TmpC=[];
  for j=1:size(x,1),
    if(z(j)==1)
      TmpB=[TmpB;x(j),y(j)]; TmpC(y(j),1)=1;
    end
  end
  TmpA=[];
  for j=1:size(TmpC,1),
    TmpA{j,1}=[];
  end
  for j=1:size(TmpB,1),
    TmpA{TmpB(j,1),1}=[TmpA{TmpB(j,1),1},TmpB(j,2)];
  end
  Tmp=Tmp(1,4); TmpB=Tmp;
  TmpC=sparse(Tmp,1,1,vn,1); count=c{i,1}-1;
  while(count>0),
    if(~(TmpC(TmpA{Tmp}(1),1)))
      Tmp=TmpA{Tmp}(1); TmpB=[TmpB,Tmp]; TmpC(Tmp,1)=1;
    else
      Tmp=TmpA{Tmp}(2); TmpB=[TmpB,Tmp]; TmpC(Tmp,1)=1;
    end
    count=count-1;
  end
  c{i,3}=TmpB;
end
for i=1:cn,
  Tmp=[0,0,0];
  for j=1:c{i,1},
    Tmp=Tmp+v(c{i,2}(j),:);
  end
  Tmp=Tmp/c{i,1}; c{i,4}=Tmp;
end
Tmp=sqrt(a*a+b*b+cc*cc);
u=[a/Tmp,b/Tmp,cc/Tmp]; uzp=u; ux=[1,0,0]; Tmp=cross(u,ux);
TmpA=sqrt(Tmp(1)*Tmp(1)+Tmp(2)*Tmp(2)+Tmp(3)*Tmp(3));
uyp=Tmp/TmpA; uxp=cross(uyp,uzp);
R=[uxp,0;uyp,0;uzp,0;0,0,0,1];
vp=(R*[v';ones(1,vn)])'; vp=vp(:,1:2); ad=min(vp,[],1);
vp=vp-[ad(1)*ones(vn,1),ad(2)*ones(vn,1)]; cs=[];
for i=1:cn,
  Tmp=R*[c{i,4}';1]; Tmp=Tmp(1:2)'-ad;
  c{i,5}=Tmp; cs=[cs;Tmp];
end
LB=min(vp(:,1)); UB=max(vp(:,1)); Tmp=UB-LB;
LBv=LB+0.1*Tmp; UBv=UB-LBv; Tmp=min(cs(:,1));
LB=min(cs(:,1)); UB=max(cs(:,1)); Tmp=UB-LB;
LBc=LB+0.1*Tmp; UBc=UB-LBc;
% cell
cvm=sparse(cn,vn);
for i=1:cn,
  for j=1:c{i,1},
    cvm(i,c{i,2}(j))=1;
  end
end
nghc=sparse(cn,cn);
for i=1:(cn-1),
  for j=(i+1):cn,
    Tmp=find(cvm(i,:) & cvm(j,:));
    if(~isempty(Tmp))
      TmpN=size(Tmp,2);
      if(TmpN>1)
        for k=1:TmpN,
          nghc(i,j)=1; nghc(j,i)=1;
        end
      end
    end
  end
end
N=cn; LMat=sparse(1,N); UMat=sparse(1,N);
for i=1:cn,
  if(cs(i,1)<=LBc)
    LMat(1,i)=1;
  end
  if(cs(i,1)>=UBc)
    UMat(1,i)=1;
  end
end
NeMat=nghc; Blocked=randperm(N);
% bond
[p,q,r]=find(triu(nghc));
b=[p,q]; bn=size(b,1); bcm=sparse(bn,bn);
for i=1:bn,
  bcm(i,b(i,1))=1; bcm(i,b(i,2))=1;
end
nghb=sparse(bn,bn);
for i=1:cn,
  Tmp=find(bcm(:,i));
  if(~isempty(Tmp))
    TmpN=size(Tmp,1);
    for j=1:(TmpN-1),
      for k=(j+1):TmpN,
        nghb(Tmp(j),Tmp(k))=1; nghb(Tmp(k),Tmp(j))=1;
      end
    end
  end
end
N=bn; LMat=sparse(1,N); UMat=sparse(1,N);
for i=1:bn,
  if((cs(b(i,1),1)<=LBc) | (cs(b(i,2),1)<=LBc))
    LMat(1,i)=1;
  end
  if((cs(b(i,1),1)>=UBc) | (cs(b(i,2),1)>=UBc))
    UMat(1,i)=1;
  end
end
NeMat=nghb; Blocked=randperm(N);
% vertice
nghv=sparse(vn,vn);
for i=1:cn,
  Tmp=[c{i,3},c{i,3}(1)];
  for j=1:c{i,1},
    nghv(Tmp(j),Tmp(j+1))=1; nghv(Tmp(j+1),Tmp(j))=1;
  end
end
LMat=sparse(1,vn); UMat=sparse(1,vn);
for i=1:vn,
  if(vp(i,1)<=LBv)
    LMat(1,i)=1;
  end
  if(vp(i,1)>=UBv)
    UMat(1,i)=1;
  end
end
N=vn; NeMat=nghv; Blocked=randperm(N);
% edge
[p,q,r]=find(triu(nghv));
e=[p,q]; en=size(e,1); evm=sparse(en,en);
for i=1:en,
  evm(i,e(i,1))=1; evm(i,e(i,2))=1;
end
nghe=sparse(en,en);
for i=1:vn,
  Tmp=find(evm(:,i));
  if(~isempty(Tmp))
    TmpN=size(Tmp,1);
    for j=1:(TmpN-1),
      for k=(j+1):TmpN,
        nghe(Tmp(j),Tmp(k))=1; nghe(Tmp(k),Tmp(j))=1;
      end
    end
  end
end
N=en; LMat=sparse(1,N); UMat=sparse(1,N);
for i=1:en,
  if((vp(e(i,1),1)<=LBc) | (vp(e(i,2),1)<=LBc))
    LMat(1,i)=1;
  end
  if((vp(e(i,1),1)>=UBc) | (vp(e(i,2),1)>=UBc))
    UMat(1,i)=1;
  end
end
NeMat=nghe; Blocked=randperm(N);
```

Number of vertices

```
% numofvertices.m
clear all; dimmin=2; dimmax=9; batches=5;
dvn=[]; cpu=[]; nmax=1000; rand('state',sum(100*clock));
for i=dimmin:dimmax,
  for j=1:batches,
    n=round(nmax/i); x=rand(n,i); t=cputime;
    [v,c]=voronoin(x); cpu(i,j)=(cputime-t)/n;
    lend=~floor(v); hend=~(ceil(v)-ones(size(v)));
    lhend=lend & hend; in=min(lhend,[],2);
    dvn(i,j)=sum(in)/n;
  end
end
dvn=[(1:dimmax)',dvn]; dvn=dvn(2:dimmax,:);
figure(1); clf;
for i=1:batches,
    semilogy(dvn(:,1),dvn(:,(i+1)),'.','LineWidth',2);
    hold on;
end
edvn=[dvn(:,1),sum(dvn(:,2:(batches+1)),2)/batches];
tmp=edvn(:,2)./exp(edvn(:,1));
A=sum(tmp)/(dimmax-1); m=[dimmin,dimmax];
semilogy(m,A*exp(m));
cpu=[(1:dimmax)',cpu]; cpu=cpu(2:dimmax,:); figure(2); clf;
for i=1:batches,
    semilogy(cpu(:,1),cpu(:,(i+1)),'.','LineWidth',2); hold on;
end
ecpu=[cpu(:,1),sum(cpu(:,2:(batches+1)),2)/batches];
tmp=ecpu(:,2)./exp(ecpu(:,1));
B=sum(tmp)/(dimmax-1); m=[dimmin,dimmax];
semilogy(m,(B/35)*(exp(1)+2).^m);
```

Vertices per cell and cell ratio

```
% numveachcell.m
clear all; dimmin=2; dimmax=6; batches=5; nmax=3000;
rand('state',sum(100*clock));
for i=dimmin:dimmax,
  for j=1:batches,
    n=round(nmax*2/i); x=rand(n,i); [v,c]=voronoin(x);
    fleet{i,j,1}=v; fleet{i,j,2}=c; fleet{i,j,3}=n;
  end
end
for i=dimmin:dimmax,
  for j=1:batches,
    v=fleet{i,j,1}; c=fleet{i,j,2}; n=fleet{i,j,3};
    lend=~floor(v); hend=~(ceil(v)-ones(size(v)));
    lhend=lend & hend; in=min(lhend,[],2);
    numvc=[]; vcin=[];
    for p=1:n,
      numvc=[numvc,size(c{p},2)]; flag=1;
      for q=1:numvc(p),
        if(~in(c{p}(q)))
          flag=0; break;
        end
      end
      if(flag)
        vcin=[vcin,numvc(p)];
      end
    end
    tmpn=size(vcin,2);
    rcin(i,j)=tmpn/n; vec(i,j)=sum(vcin)/tmpn;
  end
end
dum=rcin; str={'c_{in} / c_{all}'};
dum=dum(2:dimmax,:); tmp=[];
for i=dimmin:dimmax,
  tmp=[tmp;i*ones(batches,1),dum((i-1),:)'];
end
figure(1);
semilogy(tmp(:,1),tmp(:,2),'.','LineWidth',2); hold on;
[p,s,mu]=polyfit(tmp(:,1),tmp(:,2),4);
x=(dimmin:.02:dimmax)'; y=polyval(p,x,[],mu); semilogy(x,y);
dum=vec; dum=dum(2:dimmax,:); tmp=[];
for i=dimmin:dimmax,
  tmp=[tmp;i*ones(batches,1),dum((i-1),dimmin:dimmax)'];
end
figure(2);
semilogy(tmp(:,1),tmp(:,2),'.','LineWidth',2); hold on;
edum=[dum(:,1),sum(dum(:,2:(batches+1)),2)/batches];
tmp=edum(:,2)./exp(edum(:,1));
A=sum(tmp)/(dimmax-1); m=[dimmin,dimmax];
semilogy(m,(A/70)*(exp(1)+4).^m);
xlabel('Dimension','FontSize',14); ylabel(str,'FontSize',14);
```

Face statistics in n dimensions

```
% statsgenn.m    by K N J Tiyapan, 1st July, 2001
echo off; clear all; format short g; more off;
pt1 =fopen('./v50.dat','r'); sc1 =fscanf(pt1, '%d', 4);
Dimension=sc1(1,1);  NumVAll =sc1(2,1); NumC =sc1(3,1);
sc2 =fscanf(pt1, '%f', [Dimension, NumVAll]);
VerticeAll =sc2'; CVMat =sparse(NumC, NumVAll);
CFrame =ones(NumC, 1); VCFrame=zeros(NumVAll,1);
VFrame=ones(NumVAll,1);
for i=1:NumC,
  sc1 =fscanf(pt1, '%d', 1);
  for j=1:sc1,
    sc2 =fscanf(pt1, '%d', 1);
    Num =sc2+1; CVMat(i,Num) =1;
    if ( max(abs(VerticeAll(Num, :))) > 0.5 )
      CFrame(i,1) =0; VFrame(Num,1)=0;
    end
  end
end
fclose(pt1);
for i=1:NumC,
  VInC=find(CVMat(i,:)'); NumVInC=size(VInC,1);
  if(CFrame(i,1)==1)
    for j=1:NumVInC,
      VCFrame(VInC(j,1),1)=1;
    end
  end
end
CVNiceCMat=[];
for i=1:NumC,
  if(CFrame(i,1)==1)
```

```
    CVNiceCMat=[CVNiceCMat;CVMat(i,:)];
  end
end
CNumVNiceCMat=sum(CVNiceCMat,2); NumV=sum(VCFrame);
VVCFrameMat=zeros(NumV,2);
Vertice=zeros(NumV,Dimension); Count=0;
for i=1:NumVAll,
  if(VCFrame(i,1)==1)
    Count=Count+1;
    VVCFrameMat(Count,1)=i; VVCFrameMat(Count,2)=Count;
    Vertice(Count,:)=VerticeAll(i,:);
  end
end
pt3=fopen('./n50.dat','w'); pt2=fopen('./c50.dat','r');
line=fgetl(pt2);
sc1 =fscanf(pt2, '%d', 1);
sc2=fscanf(pt2,'%f',[Dimension,NumC]); Cell=sc2';
fclose(pt2); CNeighCCMat=sparse(NumC, NumC); t=cputime;
FVAllMat=[]; FNumVAllMat=[];
for i=1:(NumC-1),
  for j=(i+1):NumC,
    VShared=and(CVMat(i,:), CVMat(j,:));
    NumShared =sum(VShared, 2);
    NumFVAllMat=size(FVAllMat,1);
    if (NumShared >= Dimension)
      CNeighCCMat(i,j) =1; CNeighCCMat(j,i) =1; Exist=0;
      for k=1:NumFVAllMat,
        MatchExistingFV=sum(and(VShared,FVAllMat(k,:)),2);
        if(MatchExistingFV>=Dimension)
          Exist=1; break;
        end
      end
      if(Exist==0)
        FVAllMat=[FVAllMat;VShared];
        FNumVAllMat=[FNumVAllMat;NumShared];
      end
    end
  end
end
FVMat=[]; FNumVMat=[]; FVCFMat=[]; FNumVCFMat=[];
for i=1:NumFVAllMat,
  VThisFace=find(FVAllMat(i,:)');
  NumVThisFace=size(VThisFace,1);
  IncludeMe=1; IncludeMeToo=1;
  for j=1:NumVThisFace,
    if(VFrame(VThisFace(j,1),1)==0)
      IncludeMe=0;
    end
    if(VCFrame(VThisFace(j,1),1)==0)
      IncludeMeToo=0;
    end
  end
  if(IncludeMe==1)
    FVMat=[FVMat;FVAllMat(i,:)];
    FNumVMat=[FNumVMat;FNumVAllMat(i,:)];
  end
  if(IncludeMeToo==1)
    FVCFMat=[FVCFMat;FVAllMat(i,:)];
    FNumVCFMat=[FNumVCFMat;FNumVAllMat(i,:)];
  end
end
NumFVMat=size(FVMat,1); NumFVCFMat=size(FVCFMat,1);
FDim=Dimension-1;
```

```
fprintf(pt3,'Face dimension: %d\n',FDim);
fprintf(pt3,'Number of faces: %d\n',NumFVMat);
fprintf(pt3,'Number of vertices: \n [');
for i=1:NumFVMat,
  fprintf(pt3,'%d ',FNumVMat(i,1));
  if(mod(i,10)==0)
    fprintf(pt3,'...\n');
  end
end
fprintf(pt3,']\n');
fprintf(pt3,'Number of faces of nice cells: %d\n',NumFVCFMat);
fprintf(pt3,'Number of vertices: \n [');
for i=1:NumFVCFMat,
  fprintf(pt3,'%d ',FNumVCFMat(i,1));
  if(mod(i,10)==0)
    fprintf(pt3,'...\n');
  end
end
fprintf(pt3,']\n'); DVMat=FVCFMat; NumD=NumFVCFMat;
for d=3:Dimension,
  FaceCond=Dimension-d+2; DNeighDDMat=sparse(NumD,NumD);
  dVMat=[]; dNumVMat=[];
  for i=1:(NumD-1),
    for j=(i+1):NumD,
      VShared=and(DVMat(i,:), DVMat(j,:));
      NumShared =sum(VShared, 2); NumdVMat=size(dVMat,1);
      if (NumShared >= FaceCond)
        DNeighDDMat(i,j) =1; DNeighDDMat(j,i) =1; Exist=0;
        for k=1:NumdVMat,
          MatchExistingdV=sum(and(VShared,dVMat(k,:)),2);
          if(MatchExistingdV>=FaceCond)
            Exist=1; break;
          end
        end
        if(Exist==0)
          dVMat=[dVMat;VShared]; dNumVMat=[dNumVMat;NumShared];
        end
      end
    end
  end
  FDim=Dimension-d+1; fprintf(pt3,'Face dimension: %d\n',FDim);
  fprintf(pt3,'Number of faces: %d\n',NumdVMat);
  if(FDim~=1)
    fprintf(pt3,'Number of vertices: \n [');
    for i=1:NumdVMat,
      fprintf(pt3,'%d ',dNumVMat(i,1));
      if(mod(i,10)==0)
        fprintf(pt3,'...\n');
      end
    end
    fprintf(pt3,']\n');
  end
  DVMat=dVMat; NumD=NumdVMat;
  if(FDim==2)
    FVMat=DVMat;
  end
end
Time=cputime-t; NumNiceC=sum(CFrame); NumVBound=sum(VFrame);
```

Bibliography

George O. Abell. The distribution of rich clusters of galaxies. *The*

Astrophysical Journal. Supplement Series. **3**, Supplement Number 31, 211–288. May, 1958.

D. A. Aboav and T. G. Langdon. The shape of grains in a polycrystal. *Metallography.* **2**, 171–178. 1969.

D. A. Aboav. The arrangement of grains in a polycrystal. *Metallography.* **3**, 383–390. 1970.

D. A. Aboav. The stability of grains in a polycrystal. *Metallography.* **4**, 425–441. 1971.

D. A. Aboav. The arrangement of cells in a net. *Metallography.* **13**, 43–58. 1980.

Joan Adler and Amnon Aharony. Diffusion percolation. I. Infinite time limit and bootstrap percolation. *Journal of Physics A. Mathematical and General.* **21**, 1387–1404. 1988.

P. K Ahnelt., E. Fernandez, O. Martinez, J. A. Bolea, A. Kubber-Heiss. Irregular S-cone mosaics in felid retinas. Spatial interaction with axonless horizontal cells, revealed by cross correlation. *Journal of the Optical Society of America. A, Optics, Image Science, and Vision.* **17**, 3, Mar. 2000, p. 580–8.

John Algeo. *Problems in the origins and development of the English language.* 3rd ed. Harcourt Brace Jovanovich. 1966 (1982).

J. Ammermuller, W. Mockel, P. Rujan. A geometrical description of horizontal cell networks in the turtle retina. *Brain Research.* **616**, 1–2, 9th Jul. 1993, p. 351–6.

Alizabeth Andrews. *Healthy practice for musicians.* Rhinegold. London. 1997.

Francis Bacon. *Novum Organum.* (The new organon. or true directions concerning the interpretation of nature.) 1620. Translation. James Spedding, et al. Taggard and Thompson. 1863.

C. Bradford Barber, David P. Dopkin and Hannu Huhdanpaa. The Quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software.* vol. 22, no. 4, 469–483. December, 1996.

Claude Berge. *The theory of graphs and its applications.* 1958. Translation. Alison Doig. Methuen. Wiley. 1962(1964).

Gavin Betts. *Latin.* Teach Yourself Books. 2000.

Citr Bhụmiśakd́xi. *The Lahu or Mụzọe̱r language.* [Bạṣạ Lahu rhụe Mụzọeṙ] Maif.ngạm. Thailand. 1963. (in Thai.)

G. Birkhoff and S. M. Lane. *A survey of modern algebra.* A K Peters. 1997.

Reginald Horace Blyth. *Oriental Humour.* Hokuseido. Tokyo. 1959.

Cao Bohan. The Chinese language movement since the May Fourth Period. *in Language reform in China.* Peter J. Seybolt and Gregory Kuei-ke Chiang. Eds. M. E. Sharp Inc. 1978 (1979).

A. Bravais. *A. Bravais' Abhandlungen über symmetrische polyeder.* (1849). Ostwald's Klassiker der Exakten Wissenschaften. no. 17, 46–49. Wilhelm Engelmann. Leipzig. 1890.

S. R. Broadbent and J. M. Hammersley. Percolation processes I, Crystals and mazes. *Proceedings of the Cambridge Philosophical Society.* 54, 629–641. 1957.

Ronald S. Burt. *Structural holes. The social structures of competition.* Harvard University Press. 1992.

A. G. W. Cameron. Abundance of the elements in the solar system. *Space Science Reviews.* 15, 121–147. 1973.

J. Chalupa, P. L. Leath and G. R. Reich. Bootstrap percolation on a Bethe lattice. *Journal of Physics C, Solid State Physics.* **12**, L31–L35. 1979.

Sen Chung Chan. *Studies of separation of dilute dispensions.* Ph.D. thesis. UMIST, Manchester, U.K. 1990.

K. Clarkson and P. Shor. Applications of random sampling in computational geometry, II. *Discrete Computational Geometry.* **4**, 387–421. 1989.

W. D. Clayton. Studies in the *Gramineae*. XXI. *Coelorhachis* and *Rhytachne*. A study in numerical taxonomy. *Kew Bulletin.* **24**, 309–314. 1970.

W. D. Clayton. Studies in the *Gramineae*. XXVI. Numerical taxonomy of the *Arundinelleae*. *Kew Bulletin.* **26**, 111–123. 1972.

Collins USA *Collins Road Atlas USA. Canada and Mexico.* Rand McNally. 1999.

Ken Croswell. *The alchemy of heaven.* Oxford. 1995.

H. M. Cundy and A. P. Rollett. *Mathematical models.* 3^{rd} edition, 1989. Tarquin.

C. A. Curcio, K. R. Sloan. Packing geometry of human cone photoreceptors, variation with eccentricity and evidence for local anisotropy. *Visual Neuroscience.* Vol. 9, No. 2, Aug. 1992, p. 169–80.

James Dwight Dana and Edward Salisbury Dana. *Dana's new mineralogy.* 8^{th} ed. Wiley. 1997.

P. Dean. A new Monte Carlo method for percolation problems on a lattice. *Proceedings of the Cambridge Philosophical Society.* **59**, 397–410. 1963.

P. G. de Gennes, P. Lafore and J. P. Millot. Amas accidentels dans les solutions solides désordonnées. *Journal of Physics and Chemistry of Solids.* **11**, 105–110. 1959.

Michael de Podesta. *Understanding the properties of matter.* 2^{nd} ed. Taylor & Francis. 1996 (2002).

Renati Des-cartes *Principia philosophiæ.* 1644. Ludovicum Elzevirium. *cf* René Descartes. Œuvres philosophiques. Vol.III. 1973. Garnier Frères. Ferdinard Alquié ed.

G. de Vancouleurs. The extragalactic distance scale VIII. A comparison of distance scales. *The Astrophysical Journal.* **415**, 10–32. 20^{th} September 1993 .

Charles Dickens. *A child's history of England.* 1851(1868).

G. Lejeune Dirichlet. Über die Reduction der positiven quadratischen Formen mit drei unbestimmten ganzen Zahlen. *Crelle.* Journal für die reine und angewandte Mathematik. Bd. 40, 209–227. 1850. *cf* Tiyapan (2001)

Ian L. Dryden and Kanti V. Mardia. *Statistical shape analysis.* John Wiley & Sons. 2002.

J. W. Essam. Percolation theory. *Reports on Progress in Physics.* vol. 43, 835–912, 1980.

Arthur Eugene Fitzgerald, Charles Kingsley Jr. and Alexander Kusko. *Electric Machinery.* McGraw-Hill Kogakusha. 3rd Ed. 1971.

James P. Evans and Frederick M. Chester. Fluid-rock interaction in faults of the San Andreas system. Inferences from San Gabriel fault rock geochemistry and micro structures. *Journal of Geophysical Research.* **100**, B7, 13007–13020. 10 July 1995.

Rachel Farmer. *Beginner's Russian. An easy introduction.* Teach Yourself Books. 1996.

H. L. Frisch, E. Sonnenblick, V. A. Vyssotsky, and J. M. Hammersley. Critical percolation probabilities (site problem). *Physical Review.* vol. 124, no. 4, 1021–1022. 15^{th} November 1961 .

Mason Florence, Marisa Gierlich and Andrew Dean Nystrom. *Rocky Mountains. Colorado, Wyoming, Montana and Idaho.* Lonely Planet. 3rd ed. 2001.

E. M. Forster. *Howards End.* 1910.

J. Garside and J. W. Mullin. The crystallization of aluminium potassium sulphate, a study in the assessment of crystallizer design data. Part III. Growth and dissolution rates. *Transactions of the Institution of Chemical Engineers.* vol. 46, no. 1, T11–T18. 1968.

B. Gay and P. E. Preece. Matrix methods for the solution of fluid network problems, Part I–mesh methods. *Transactions of the Institution of Chemical Engineers.* vol. 53, no. 1, 12–15. January 1975.

J. W. Giles, C. Hanson and J. G. Marsland. Drop size distributions in agitated liquid-liquid systems with simultaneous interface mass transfer and chemical reaction. in Proceedings of the International Solvent Extraction Conference, ISEC 71. vol. 1, 94–111. 1971.

A. S. Glassner. Ed. *Graphics gems.* Academic Press. 1990.

Nigel Gotteri and Joanna Michalak-Gray. *Polish.* Teach Yourself Books. 1997.

P. J. Green and R. Sibson. Computing Dirichlet tessellations in the plane. *The Computer Journal.* vol. 21, February to November, 168–173, 1978.

G. Grimmett. *Percolation.* Second edition. Springer. 1999.

Patrick Grosfils, Jean Pierre Boon, E. G. D. Cohen and L. A. Bunimovich. Propagation and organization in lattice random media. *Journal of Statistical Physics.* vol. 97, nos. 3/4, 575–608, 1999.

Branko Grünbaulm. *Convex polytopes.* Inter Science, Wiley. Pure and applied mathematics series, vol. XVI. 1967,

Branko Grünbaum and G. C. Shepherd. *Tilings and patterns.* W. H. Freeman. N.Y. 1987.

J. M. Hammersley and W. Morton. Poor man's Monte Carlo. *Journal of the Royal Statistical Society (B).* 16, 23–38. 1954.

Allen Hammond. *Which World? scenarios for the* 21st *century.* Earthscan. 1998.

C. Hammond. *The basics of crystallography and diffraction.* Oxford. 2001.

Thomas L. Hankins. Triplets and triads. Sir William Rowan Hamilton on the metaphysics of mathematics. *Isis.* **68**, 242, 175–193. 1977.

John Happel. Viscous flow relative to arrays of cylinders. *American Institute of Chemical Engineers Journal.* vol. 5, no. 2, 174–177. June, 1959.

G. H. Hardy and E. M. Wright. *An introduction to the theory of numbers.* 1938. Oxford. or 5th ed. 1979.

H. W. Harrison and P. M. Trotman. *Foundations, basements and external works. Performance, diagnosis, maintenance, repair and the avoidance of defects.* BRE. 2002.

P. M. Heertjes. Filtration. *The Transactions of the Institution of Chemical Engineers.* vol. 42, no. 7, T266–T274, 1964.

H. S. Hele-Shaw. The flow of water. *Nature.* Vol. 58, May 12, 1898, p. 34–36.

H. S. Hele-Shaw. The motion of a perfect liquid. *Nature.* Vol. 60, September 7, 1899, p. 446–451.

Franz E. Hohn. *Elementary matrix algebra.* 3rd ed. Macmillan. 1958 (1973).

D. Houi and R. Lenormand. Experimental and theoretical study of particle accumulation at the surface of a filter. in the Proceedings of 4th World Filtration Contress. no. 1, 1.1–1.6. 1986.

Fred Hoyle. *Ten faces of the universe.* Heinemann. London. 1977.

Fred Hoyle and Chandra Wickramasinghe. *Life cloud* the origin of life in the universe.

J. M. Dent & Sons. 1978.: Fred Hoyle and Nalin Chandra Wickramasinghe. *Diseases from space.* J. M. Dent & Sons. 1979.

Charlie Ireland. Private communication. from Invercargill, New Zealand. 12th August 2002

C. Isenberg. *The science of soap films and soap bubbles.* Tieto. 1978.

K. J. Ives. Deep bed filtration. in Solid–Liquid Separation. 2nd. Butterworths Monographs in Chemistry and Chemical Engineering, Ladislav Svarovsky. ed. 284–301. 1977.

N. M. Jackson. *A mathematical model to simulate the structure and performance of porous media.* Thesis. UMIST, Manchester, 1994.

N. M. Jackson, R. Jafferali, D. J. Bell and G. A. Davies. A study of the structure of micro and ultra filtration membranes. The Voronoi tessellation as a stochastic model to simulate the structure. *Journal of Membrane Science.* **162**, 23–43. 1999.

Riaz Jafferali. *A stochastic model to simulate the structure and performance of microfiltration media and the growth of animal cell cultures.* Thesis. UMIST, Manchester, 1995.

Harold James. *The German slump. Politics and economics 1924–1936.* Clarendon. Oxford. 1986(1987).

Vera Javarek and Miroslava Sudjić *Serbo-Croat.* 2nd ed. Teach Yourself Books. 1963(1972).

G. R. Jerauld, J. C. Hatfield, L. E. Scriven and H. T. Davis. Percolation and conduction on Voronoi and triangular networks. A case study in topological disorder. *Journal of Physics C. Solid State Physics.* **17**, 1519–1529. 1984.

A. J. Jones. *Crystallization process sytems.* Butterworth Heinemann. 2002.

Immanuel Kant. Beantwortung der Frage. Was ist Aufklärung. *Berlinische Monatsschrift.* **4**. December. 1784. 481–494.

Alan R. Kerstein. Equivalence of the void percolation problem for overlapping spheres and a network problem. *Journal of Physics A. Mathematical and General.* **16**, 3071–3075. 1983.

Daniel Keyes. *The minds of Billy Milligan.* Bantam. 1999.

Hassan K. Khalil. *Nonlinear Systems.* Prentice Hall. 2nd ed. 1996.

Amnad Khitapanna. (Aṃṇạc Gịtȧbarṇạ) personal communication. 2000.

C. Klein and C. S. Hurlbut, Jr. *Manual of mineralogy.* 21st Edition. Wiley. 1993.

D. E. Knuth. *The TEX.* Addison Wesley. 1970.

J. W. de Kock and M. R. Judd. The flow pattern within a void in a porous mass. *The Transactions of the Institution of Chemical Engineers.* vol. 43, no. 3, T78–T84. 1965.

J. M. Kosterlitz and D. J. Thouless. Ordering, metastability and phase transitions in two-dimensional systems. *Journal of Physics C. Solid State Physics.* vol. 6, 1181–1203.

John Krige. *Science, revolution and discontinuity.* The Harvester Press. 1980.

Roderic Lakes. Foam structures with a negative Poisson's ratio. *Science.* **235**, 1038–1040. 1987.

A. Levy, S. Reich and P. Meakin. The shape of clusters on rectangular 2D lattices in a simple "phase separation" computer experiment. *Physics Letters.* vol. 87A, no. 5, 248–252. 11th January 1982 .

Larry S. Liebovitch and Daniela Scheurle. Two lessons from fractals to chaos. *Complexity.* vol. 5, no. 4, 34–43. March-April, 2000.

K. C. Lim, M. A. Hashim and B. Sen Gupta. The effect of volume shape factor on the crystal size distribution of fragments due to attrition. *Crystal Research Technology.* vol. 34, no. 4, 491–502. 1999.

J. Litwiniszyn. Colmatage considered as a certain stochastic process. *Bulletin de L'Academie Polonaise des Sciences.* vol. XI, no. 3, 117–121. 1963.

J. Litwiniszyn. On some mathematical models of the suspension flow in porous medium. *Chemical Engineering Science.* vol. 22, 1315–1324. 1967.

C. W. Lowe. Some techniques of evolutionary operation. *Transactions of the Institution of Chemical Engineers.* vol. 42, no. 9, T334–T344. 1964.

Ian S. Lustick and Dan Miodownik. Deliberative democracy and public discourse. The agent-based argument repertoire model. *Complexity.* vol. 5, no. 4, 13–30. March-April, 2000.

T. Magay and L. Országh. *A concise Hungarian–English dictionary.* Oxford University Press. Akademiai Kiadó 1981 (1990).

K. R. Mecke and A. Seyfried. Strong dependence of percolation thresholds on polydispersity. *Europhysics Letters.* **58** (1), 28–34, 1st April 2002 .

Hans Meinhardt. *The algorithmic beauty of sea shells.* Springer-Verlag. 1995 (1998).

Stephen A. Miler and Amos Nur. Permeability as a toggle switch in fluid-controlled crustal processes. *Earth and Planetary Science Letters.* **183**, 133–146. 2000.

R. E. Miles. A random division of space. *Special Supplement to Advances in Applied Probability.* 243–266. 1972.

Gian Maria Milesi-Ferretti and Assaf Razin. Current account reversals and currency crises. Empirical regularities. in *Currency Crisis.* Paul Krugman. ed. University of Chicago Press. 2000.

U. Mizutani. *Introduction to the electron theory of metals.* Cambridge. 2001.

K. K. Mohanty, J. M. Ottino and H. T. Davis. Reaction and transport in disordered composite media. Introduction of percolation concepts. *Chemical Engineering Science.* vol. 37, no. 6, 905–924. 1982.

J. Møller. Random tessellations in $\Re^d$. *Advances in Applied Probability.* vol. 21, p. 37–73, 1989.

M. Mulder. *Basic principles of membrane technology.* Kluwer. 1992.

T. Mulder and R. Gimbel. On the development of high performance filtration materials for deep bed filters. in proceedings for the 5$^{\text{th}}$ World Filtration Congress. vol. 3, 45–57. Nice, 1990.

Andrew Nathaniel Nelson. *Japanese–English Character Dictionary.* Charles E. Tuttle. 1962.

James Naughton. *Colloquial Slovak.* 1997.

Johann Nittmann, Gérard Daccord and H. Eugene Stanley. Fractal growth of viscous fingers, quantitative characterization of a fluid instability phenomenon. *Nature* Vol. 314, 14 March 1985, p. 141–144

A. Okabe, B. Boots and K. Sugihara. *Spatial tessellations concepts and applications of Voronoi diagrams.* Wiley. 1992.

I. M. Oshanina. *Chinese–Russian Dictionary.* [Kitaĭsko–Russkiĭ Slovarb] State Publisher. Foreign and National Lexicographer. [Gosudarstvennoe Izdatelbstvo. Inostrannyĭ i Nacionalbnyĭ Slovareĭ] Moscow. 1955.

R. K. Pathria. *Statistical Mechanics.* 2nd ed. Butterworth-Heinemann. 1996 (1997).

Roger Penrose. *The emperor's new mind. Concerning computers, minds, and the law of physics.* 1989.

R. Penrose. Tilings and quasi-crystals. a non-local growth problem? in *Introduction to the mathematics of quasicrystals.* Marko V. Jarić Ed. 53–79. Aperiodicity and order. Vol. 2. Academic Press. 1989.

Petersburg *Investment projects of St. Petersburg* Information and Consulting Centre. International Finance and Enterprise Week. Manchester International Conference Centre. 31st October–2nd November 2001 .

F. C. Phillips. *An introduction to crystallography.* Longmans. 3rd ed. 1949 (1963).

H. W. Piekaar and L. A. Clarenburg. Aerosol filters – the tortuosity factor in fibrous filters. *Chemical Engineering Science* vol. 22, 1817–1827. 1967.

Plato. [Aristocles] *Phaedo.* [Phaidon] 360 BC.

Michael de Podesta. *Understanding the properties of matter.* Taylor & Francis. 2nd ed. 2002.

J. B. Poole and D. Doyle. Research in solid-liquid separation. *The Chemical Engineer.* no. 190, CE169–CE172. July–August, 1965.

K. E. Porter. Liquid flow in packed columns. Part I. The rivulet model. *The Transactions of the Institution of Chemical Engineers.* vol. 46, no. 3, T69–T73. 1968.

J. G. Powles and N. Quirke. Fractal geometry and Brownian motion. A new parameter to describe molecular motion. *Physical Review Letters.* Vol. 52, No. 18, 1571–1574. 30 April 1984.

Burkhard A. Prause. *Magnetic resonance imaging of structure and coarsening in three-dimensional foams.* Ph.D. Dissertation, the University of Notre Dame. 2000.

Franco P. Preparata and Michael Ian Shamos. *Computational Geometry. An introduction.* Springer-Verlag. 1985.

Przemyslaw Prusinkiewicz and Aristid Lindenmayer. *The algorithmic beauty of plants.* Springer-Verlag. 1990.

L. R. Pujara and Naresh Shanbhag. Some stability theorems for polygons of polynomials. *IEEE Transactions on Automatic Control.* Vol. 37. No. 11. Nov. 1992. 1845–1849

L. Rai. Pujara. On the pseudoboundary of unstable polytopes of polynomials. *IEEE Transactions on Automatic Control.* **41**, 8. 1188–1190. August 1996.

Annick van Put, Akos Vertes, Darek Wegrzynek, Boris Treiger and René van Grieken. Quantitative characterization of individual particle surfaces by fractal analysis of scanning electron microscope images. *Fresenius Journal Analytical Chemistry.* **350**, 440–447, 1994.

J. H. Raistrick. The nature of adsorptive filtration. in Proceedings of the 4th World Filtration Congress. R. Vanbrabant, J. Hermia and R. A. Weiler, Eds. Part I. p. 1.65–1.72. 1986.

N. Rivier. Recent results on the ideal structure of glasses. *Journal de Physique.* Colloque C9, Supplément au no. 12, **43**, C9-91–C9-95. December, 1982.

Boyd Robertson and Iain Taylor. *Gaelic. A complete course for beginners.* Teach Yourself Books. 1993.

Steven Rodelet and Jeffrey Sacs. The onset of the East Asian Financial Crisis. in *Currency Crisis.* Paul Krugman. ed. University of Chicago Press. 2000.

H. E. Rose and J. E. English. The influence of blinding material on the results of test sieving. *Transactions of the Institution of Chemical Engineers.* vol. 51, 14–21. 1973.

P. N. Rowe and R. Collins. The flow pattern within a void in a porous mass, *The Transactions of The Institution of Chemical Engineers.* vol. 43, no. 7, T217–T220. 1965.

P. N. Rowe and C. Yacono. The distribution of bubble size in gas fluidised beds. *Transactions of the Institution of Chemical Engineers.* vol. 53, no. 1, 59–60. January, 1975.

G. S. Rushbrooke and D. J. Morgan. On the magnetically dilute Heisenberg and Ising ferromagnetics. *Molecular Physics.* vol. 4, 1–15, 1961.

Joan Russell. *Swahili.* Teach Yourself Books. 1996.

Oliver Sacks. *Awakenings.* Piccador. Pan Books. 1973(1982).

Oliver Sacks. *seeing voices. A journey into the world of the deaf.* Piccador. Pan Books. 1989(1990).

Muhammad Sahimi. *Applications of percolation theory.* Taylor & Francis. 1994.

George Salmon. *A treatise on the analytic geometry of three dimensions.* revised by Reginald a. P. Rogers. 5th ed. Longman, Green and Co. 1912.

Arnold Schoenberg. *Harmonielehre.* 1978. English translation. Theory of Harmony. Farber. 1983.

Ralf Schumacher. *A stochastic model to simulate the structure and performance of cellular polymeric membranes in dead end filtration.* Dipolmarbeit. [Ph.D. thesis] Institute für chemische Verfahrenstechnik der Technischen Universität Clausthal and UMIST. 1996.

Madan G. Singh. Ed. *Systems & Control Encyclopedia.* Pergamon. 1987.

John Maynard Smith and Eörs Szathmáry. *The major transitions in evolution.* W. H. Freeman. 1995.

Herbert Warington Smyth. *Five years in Siam, from 1891–1896.* J. Murray. London. 1898. in two volumes.

Rupert Snell. *Beginner's Hindi script.* Teach Yourself Books. 2000.

Murray R. Spiegel. *Probability and Statistics.* Schaum's outline. McGraw-Hill. 1975.

Dietrich Stauffer and Amnon Aharony. *Introduction to percolation theory.* 1985. 2nd ed. Taylor & Francis. 1992. and 2nd revised ed. 1994 (1998)

R. L. Stevenson. *Dr. Jenkyll and Mr. Hyde.* 1886.

Robert Louis Stevenson. *The amateur emigrant.* 1895.

William C. Stokoe Jr., Dorothy C. Casterline and Carl G. Croneberg. *A Dictionary of American Sign Language on Linguistic Principles.* Gallaudet College Press. 1965

L. Svarovsky. Characterization of particles suspended in liquids. in Solid-liquid separation. 2nd ed. L. Svarovsky, Ed. Butterworths. p. 8–32. 1977.

Masaharu Tanemura, Tohru Ogawa and Naofumi Ogita. A new algorithm for three-dimensional Voronoi tessellation. *Journal of Computational Physics.* **51**, 191–207. 1983.

Aspasia Theodosiou. personal communication. 2002.

J. R. R. Tolkien. *Lord of the Rings.* 1955.

C. R. G. Treasure. Fine particle-size classification. *Transactions of the Institution of Chemical Engineers.* vol. 43, no. 6, T199–T205. 1965.

U. of Man. *Mathematics, the University of Manchester, 2000 and 2001 entry.* booklet. the University of Manchester. 2000.

Rüdiger Vaas. Ed. Urknall für Einsteiger. *Bild der Wissenschaft.* May 2002.

G. F. Voronoï New application of continuous parameters to the theory of quadratic form. First memoir. On some properties of the perfect positive quadratic forms. *Journal für die reine und angewandte Mathematik.* **133**, 97–178, 1908. *cf* Tiyapan (2001). KNT1(i).

G. F. Voronoï New application of continuous parameters to the theory of quadratic form. Second memoir. Research on the primitive parallelohedron. *Journal für die reine und angewandte Mathematik.* **134**, 1908. *cf* Tiyapan (2001). KNT1(ii).

G. F. Voronoï New application of continuous parameters to the theory of quadratic form. Second memoir. Studies on the primitive parallelohedra. *Journal für die reine und angewandte Mathematik.* **136**, 67–181, 1909. *cf* . Tiyapan (2001).

Tuan Duc Vuong and John Moore. *Colloquial Vietnamese. A complete language course.* Routledge. 1994.

N. Walsh. *Making TEX work.* O'Reilly. 1994.

David Ward. Language cull could leave people speechless. *The Guardian.* Saturday 25th May 2002 . National news, 13.

Magnus Joseph Wenninger. *Polyhedron models.* Cambridge University Press. 1971.

R. van de Weygaert and V. Icke. Fragmenting the universe II. Voronoi vertices as Abell clusters. *Astronomy and Astrophysics.* **213** 1–9. 1989.

R. van de Weygaert. Fragmenting the universe III. The construction and statistics of 3-d Voronoi tessellation. *Astronomy and Astrophysics.* **283**, 361–406. 1994.

Angela Wilkes and John Shackell. *Welsh for beginners.* 1989.

Arnett Wilkes and Nicholias Nkosi. *Zulu. A complete course for beginners.* Teach Yourself Books. 1995.

David Wilkinson and Jorge F. Willemsen. Invasion percolation. a new form of percolation theory. *Journal of Physics A. Mathematical and General.* **16**, 3365–3376. 1983.

E. T. Wilkinson, A. R. N. Fairclough and G. A. Davies. The filtration of dilute suspensions using non-woven cloths and membranes. in Proceedings of the 4th World Filtration Congress. R. Vanbrabant, J. Hermia and R. A. Weiler, Eds. Part I. p. 1.7–1.18. 1986.

Trevor Williams and Rolf Bjerknes. Stochastic model for abnormal clone spread through epithelial basal layer. *Nature.* vol. 236, 19–21. 3rd March 1972 .

X. J. Zhan, J. B. Troy. Modeling cat retinal beta-cell arrays. *Visual Neuroscience.* Vol. 17, No. 1, Jan.–Feb. 2000, p. 23–39.

My writings, Kittisak Nui Tiyapan

1991 *Antimonytrioxide extraction from stibnum ore by hydrometallurgical method.* Senior project. [Final year project]. Mining Engineering, Chulalongkorn University, Thailand. 1991. KNT2(i).

1994 Cyberspace. *Articles Online.* ATSIST. 12th December 1994 . also at www.nectec.or.th/bureaux/atsist KNT3(i).

1995 The End of Stars. *Articles Online.* ATSIST. 20th February 1995 . also at www.nectec.or.th/bureaux/atsist KNT4(i).

1995 Self-tuning Extremum Control. *Design Exercise Report.* under the supervision of Dr. M. B. Zarrop. M.Sc. Course in Control and Information Technology 1994/1995. Control System Centre, UMIST. KNT4(ii).

1995 Computation of Fluid Flow. *M.Sc. dissertation.* under the supervision of Professor G. A. Davies and Professor D. J. Bell. September. UMIST. KNT4(iii).

1995 The story of Andromedra. *Sakkayaphab.* November. **3**, 2, 24–25. ATPIJ, Japan. Thai translation by *Sroemśakd̂xi Ụ̀atrongcit̲t̲a.* KNT4(iv).

1996 Let's start at the very beginning. *Sakkayaphab.* January. **3**, 4, 23–25. ATPIJ, Japan. Thai translation by *Sroemśakd̂xi Ụ̀atrongcit̲t̲a.* KNT5(i).

1996 To be unkempt. *Sakkayaphab.* April. **3**, 7. ATPIJ, Japan. Thai translation by *Suvanjay Bongs̲asukicvaḍhana.* KNT5(ii).

1996 I see a white car before me. *Sakkayaphab.* June. **3**, 9, 12–13. ATPIJ, Japan. Thai translation by *Taśa̲niya̲ Mẹdhạbisiṭh.* KNT5(iii).

1996 Critical probability and other properties of 2-d tessellation *Mathematical Theory of Networks and Systems–96.* 24th –28th June (27th June , Session on Analysis of Queuing Networks). The Ritz-Carlton, St. Louis, Missouri. KNT5(iv).

1996 Some properties of stochastic optimal control. submitted in July 1996 to the 7th International Symposium on Dynamic Games and Applications (16–18 December 1996). Japan. KNT5(v).

1996 On an algorithm for object-location problem. [Object location using Extremum Control] *1996 Advanced Theory and Application of Control Systems.* 6th –8th October . Izu Recreation Centre, Atagawa Heights, Atagawa, Japan. Fu-I1–Fu-I5. KNT5(vi).

1996 On pragmatists and idealists. submitted to Sakkayaphab. 21st October 1996 . KNT5(vii).

1996 [Nhạun khạung Mạur̈ris] (The Morris Worm). *Sakkayaphab.* December. **4**, 3, 20–22. ATPIJ, Japan. KNT5(viii).

1997 Fractals in traffic control. *The Fourth Annual Conference of Thai Researchers in Japan.* abstract. 23rd February . Nippon Seinenkan Hotel, Tokyo. TSAJ and ATPIJ. p. 18 KNT6(i).

1997 Distributed parameter systems. *The Fourth Annual Conference of Thai Researchers in Japan.* abstract. 23rd February . Nippon Seinenkan Hotel, Tokyo. TSAJ and ATPIJ. p. 19 KNT6(ii).

1997 Vision robots. *The Fourth Annual Conference of Thai Researchers in Japan.* abstract. 23rd February . Nippon Seinenkan Hotel, Tokyo. TSAJ and ATPIJ. p. 20 KNT6(iii).

1997 Singular perturbation. *The Fourth Annual Conference of Thai Researchers in Japan.* abstract. 23rd February . Nippon Seinenkan Hotel, Tokyo. TSAJ and ATPIJ. p. 21 KNT6(iv).

1997 Simulation techniques using RLS algorithm for object-location problem. *Proceedings of the World Congress on Systems Simulation.* 1st–3rd September . Pan Pacific Hotel, Singapore. 518–522. KNT6(v).

1997 Report for German Literature course. TIT. presented to Professor Ishikawa. 9th September 1997 . KNT6(vi).

1997 Modelling the economics. *1997 Advanced Theory and Application of Control Systems.* 10th –12th October . National Olympic Memorial

Youth Centre, Yoyogi, Shibuya-ku, Tokyo. Fu-A1–Fu-A2. KNT6(vii).

1997 Modelling economics as a flow of money within networks. submitted to the American Journal of Physics. 4th November 1997 . KNT6(viii).

1997 Modelling of traffic congestion. submitted to the Journal of Statistical Physics. 4th November 1997 . KNT6(ix).

1998 Simulation techniques using RLS algorithm for object-location problem. Paper presented to the Furuta Laboratory on an in-house seminar. TIT, Tokyo. 4-5pm. 3rd January 1998 KNT7(i).

1998 Critical probability in traffic modelling and control. Paper presented to the Furuta Laboratory on an in-house seminar. TIT, Tokyo. 2nd April 1998 . KNT7(ii).

1998 Technical report number 1. presented to Professor Katsuhisa Furuta, TIT, Japan. 16th July . KNT7(iii).

1998 Technical report number 6. On controlling the synchronous machines. presented to Professor Katsuhisa Furuta, TIT. Japan. 23rd August . KNT7(iv).

1998 Variable structure control for a singularly perturbed system. *1998 Advanced Theory and Application of Control Systems.* 26th–28th September . Hotel Ohashi, Lake Kawaguchi, Yamanashi, Japan. Fu-C1–Fu-C8. KNT7(v).

1998 Technical report number 2 (in System and Control). presented to Professor Katsuhisa Furuta, TIT. Japan. 1st November . KNT7(vi).

2000 *Interesting English. [Bhạsạ Angkṛiṣ an nạ soncaị]* Kittix. Chulalongkorn University Press. Bangkok. KNT8(i).

2000 *Free translation of English. [Plạe kleđ́ Angkṛiṣ]* Kittix. Chulalongkorn University Press. Bangkok. KNT8(ii).

2001 *The study of Voronoi tessellation.* 22nd May 2001 KNT9(i).

2001 *Voronoi Translated, Introduction to Voronoi tessellation and essays by G. L. Dirichlet and G. F. Voronoi.* Kittix. Chulalongkorn University Press. Bangkok. KNT9(ii).

Translation

G. F. Voronoi, 1909

The translation of Voronoi's paper has already appeared in a few publications, for example *Voronoi Translated* (Kit Tiyapan, 2001), *Ph.D. Thesis* (Kit Tiyapan, University of Manchester, 2004) and *Second Memoir, Studies on the Primitive Parallelohedra, Second Part, Domains of Quadratic Forms* (Kit Tyabandha translated, 2007), therefore it is omitted here.

God is Superset

Kit Tyabandha

19th June 2004

God is a mathematical definition, and the definition is that of a superset. He is the Superset, that is the biggest superset which includes not only material- but also all other things, abstract quanlities and spirits. Since God creates everything, mathematics being included, it is interesting how the Creator may be defined by the tool of His own creation.

Judaism, Christianity and Islam, all are one and the same religion. In other words they are the same believe based on the One God. All religions serve to lead one towards God. Once we have found Him, the only thing to pay attention to is Him and not the means. One may believe paths, but to believe *in* them after one has found God is polytheism. For example, before you have found God, countries serve as the mean to give unity. But this unity is ever-changing in the ever time-varying boundaries, and leaders. When one has found God, one no longer believes in any of these. A sage thus believes in no countries. A sage is a man with wisdom. And the only wisdoms are those based on the fear of God.

As God is the Superset, God-fearing is the superset of all fears. Therefore whatever you may fear are included therein. Man is but a weak, imaginary albeit imaginative creature. He can not stand, for instance, being exposed to the cosmic-ray in space. But God forges the stars and galaxies themselves, not to mention the Big Bang itself. Our physics can make neither head nor tail of this universe before Planck's time, that is 10^{-43} seconds after Big Bang. But God was already there as far as infinitely beyond that, and will always be.

Not only this, but God is also an infinite series of creators. For things not forever existing need have their beginnings and ends. Everything which has a beginning must be created. Suppose g_1 and g_2 be two different gods. Then our faith in the One argues that sometime in the past there must already existed another god G, the God, who has created both g_1 and g_2. If one say, 'Oh! But God is separated from Satan (S). He has not created the latter,' then according to this line of argument there must have been before that another god G_1 which had created both G and S. God is then the infinite series $G, G_1, G_2, \ldots, \infty$.

The Old Testament or the Talmud is a set of riddles. Therefore the Jews at the time before Christ were very unlucky because they were provided with nothing but these. Perhaps that is why there have been so many good scientists who are Jewish. According to the Old Testament it could be interpreted that the Me (M) is separated from God (G). But Jesus came and defined two things. For one thing he defined Christ (C) to be the Son of God. As G is a mathematical definition, so must C be. Christ is the Son of God. In mathematical terms, C is the subset of G, or $C \subset G$ Jesus said further that whoever believe in him he shall come and sit in his heart. Thus $M \subset C$, and therefore $M \subset C \subset G$, as repeated in words over and over in the pre-Easter parts of the New Testament, 'The Me in Christ in God.' In other words Christ is the glue which binds M to G, and makes us *no longer a slave but a son*. Then came Muhammad and said that the Me could be in God direct, without Christ being the glue. Muhammad recognises Messiah, but he also argues that $M \subset G$, that is the Me is in God. An example to confirm this can be seen in the *Seven Pillars of Wisdom* by T E Lawrence, who is normally known as Lawrence of Arabia, in the part where he talks about the Bedouin and their belief. Jesus, I think, had already known this, because he said in John:14, 'No one comes to the Father except through Me.' In his 'Lanna' pentalogy, namely *A Lanna in town*, *A Kiwi Lanna*, *A British Lanna*, *Edokko no Lanna* and *The Siamese Lanna*, Kittix, Bangkok, 2003, Tiyapan argues that this 'Me' is the reader's Me, which is the Me who believes in Christ and God. Therefore it is possible he had already said $M \subset G$ before Muhammad did. Tiyapan reasons this reader's Me is used by the author who wrote *Bhagavad Gītā* where, for example, is mentioned, 'Follow no one but Me!' Thus according to his believe, and contrary to usual lore, the latter half of *Gītā* is not the anticlimax but the climax of logical arguments compared with the other part. Gandhi in his auto-biography thinks that if Christ is the Son of God, then we all are the sons of God. Mathematically both Muhammad and Jesus had already answered this question of his earlier, as above mentioned. Buddha sought the answer to death, and found by enlightenment Truth that I think tells him one recycles, together with every other beings and things, within God. Unless Truth and God be one and the same, the former needs be nothing, as all created things are finite, and therefore nothing when juxtaposed with Infinite.

To answer some of the riddles in Old Testament, Eve is the mother of science who made man think and try to understand his Creator. Man before the Fall, as well as all the other animals, live in paradise where they are one with God and never philosophically think. The ability to think may lead to researches. But it may also lead to opinions, or *diṭṭhi* in Buddhism. Jesus says, 'I come to serve not to judge.' Buddha says likewise, 'Never have *diṭṭhi*!'

What Darwin discovered in his theory of revolution is that Creation is not an instance but a process. Tiyapan thinks this process is a percolation process. But then again he is a mathematician. More on this later. The bibles never said otherwise than this. God is a living

God, and He takes days in creating. These days of His are undefined, since it was well before the Earth and the Sun were here.

Last but by no means least Judas may be a traitor, but he is necessary. Until Jesus means in our heart no longer as a person but as the mathematical definition of Christ there can be no salvation for us. Thus when one of his best disciples told him to go to some other places where he would not be killed, he told him, 'Get behind me Satan!' At another time he told one of his most loyal disciples that after he dies that he shall renounce his name after his death.

Book Introduction

Voronoi Translated

Kit Tiyapan

2001

ISBN 974–13–1503–1

That succinct title was followed by a more descriptive one, *Introduction to Voronoi tessellation and essays by G. L. Dirichlet and G. F. Voronoi*, which summarises this book. This is the first book, and hopefully the last one, I have written which has a grey cover, *grey* being the symbol of *Gandalf the Great* which in turn rhymes with *Graham Davies* the name of my supervisor and sponsor from end 2000. It is also the first book which I typeset with TEX which I consider complicated but works, while both of the previous two books were typeset using LaTEX which I consider simple and works, the reason for the change being the upgrade of the latter to LaTEX2ε which I consider complicated and (presumeably) works together with the fact that it has TEX working in the background. The switching over thus helps reduce one shell around the equivalences of the only desirable attribute left which is that of *complicated but works*. Here the two venerable alphabets v and n rules throughout, starting from the title *Voronoi*, the name of a russian mathematician of the nineteenth century, then the picture taken in *Vienna* on the front, and that in *Venice* on the back covers; if you look carefully enough, somewhere in the book you will find another picture of a stone masonry with a caption saying *Verona* which Professor Davies decidedly says is not a Voronoi tessellation. The introduction gives some historical backgrounds and pictures of various orders of covering lattices of the Voronoi tessellation in two dimensions, which can be used to represent for example a realistic picture of the conglomeration of grains within some kind of matrix. The triangular and the hexagonal lattices are examples of dual lattices. The picture of the hexagonal and kagome lattices superimposed on top of each other is not a picture of the Eden Project in Cornwall but an example of covering lattices. This is not a free translation of the seminal works by G. F. Voronoï and G. L. Dirichlet. I tried to retain the original as much as possible, namely where I thought the grammar in the original was incorrect or the structure of the sentence is convoluted I coined up the equivalence, though this is by no mean how I always translate. As a result the translation is necessarily more difficult to read than it would have been had it been freely summarised. My reason for doing it in this style is firstly in order to transfer as much nuances across as possible, and secondly simply because I am not a mathematician and therefore in no position to summarise the work of one. Last but not least must be my own shortcoming and inexperience in translating from French and German. The preface of the book gives a brief history of the *Journal für die reine und angewandte Mathematik* while the introduction that of the Voronoi tessellation. Also one knows from the latter that the name *kagome* is a Japanese word which means *basket interstice or pattern*, the fact that I realised while reading the Japanese language in Tokyo. *Me* is the word for *eye* as well as *pattern*, and in Japanese *kago* means *an intersticed basket* as much as *taklą̀* does in Thai. There is hardly discipline in which Voronoï tessellation has not found applications and the list of relevant areas is endless, ranging from management to computation to physics. This book has 282 pages of contents.

Kit Tyabandha, Ph.D.
Manchester, England

Book Introduction
Percolation within Percolation and Voronoi Tessellation

Kit Tiyapan

2003

ISBN 974–91036–1–0

This book contains both my original thesis, as submitted to UMIST in March 2003, as well as the translation of Schumann's Liederkreis, Op. 24, from German into English and Thai that I did for Aṃnạc Giṭȧbarrṇạ who was once the president of the Bangkok Music Society. The preface says that this is not the final version of the thesis. In fact only God knows what the final thesis will look like. There are 585 pages in all, and approximately half of these are appendices. As usual I write this book and thesis in TEX. As such it is my most complex project on TEX to date. A TEX macro I wrote is given which in its packed form is over 600 lines long. It predominantly deals with how to display pictures, as well as index and cross references. Another TEX macro is also given which contains my works on transcription systems of various languages from Lanna to Chinese to ASL, that is the American Sign Language. Other programmes given are what I had used on Matlab to produce the results mentioned in the contents. The introduction and literature survey are eclectic verging on desultory, which is partly due to the multidisciplinary nature of the work. Parts of the introduction contain the contents of the emails I wrote to various people concerned when there were serious problems affecting the progress of the work. Future researchers and academicians may be able to learn from this and avoid facing or incurring them. The literature survey is distributed throughout the book instead of centralisedly staying together in one place. I give some flavour of the different disciplines concerned. These include mathematics and geometry, physics and percolation, statistics and statistical physics, tessellation and Voronoi Tessellation, quadratic forms and quadratic equations. Observing the clusters and their development is important in the study of percolation. The sudden accelerated growth of the cluster size heralds the onset of percolation. Applications of percolation theory to traffic congestion and economics transition are also mentioned. The value of percolative thresholds are obtained from the simulation for the Voronoi Tessellation, 2-homeohedral tilings, cities traffic, and n-gons and spheres in continuum. Also in the appendices are the collection of my writings both technical and nontechnical since 1994. The range of the topics of these is rather wide, for example antimony trioxide extraction, cyberspace, computer worm, economic and traffic modellings, variable structure control and singular perturbation. There are also articles on languages. And I have included translations I have made of the seminal works by Dirichlet and Voronoi on the Voronoi Tessellation.

Kit Tyabandha, Ph.D.
Manchester, England